



Commande robuste d'un effecteur par une interface cerveau machine EEG asynchrone

Alexandre Barachant

► To cite this version:

Alexandre Barachant. Commande robuste d'un effecteur par une interface cerveau machine EEG asynchrone. Traitement du signal et de l'image [eess.SP]. Université de Grenoble, 2012. Français. NNT : 2012GRENT112 . tel-01196752

HAL Id: tel-01196752

<https://theses.hal.science/tel-01196752>

Submitted on 15 Sep 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

THÈSE

POUR OBTENIR LE GRADE DE
DOCTEUR DE L'UNIVERSITÉ DE GRENOBLE
Spécialité : Signal, Image, Parole, Télécoms
Arrêté ministériel : 7 août 2006

PRÉSENTÉE PAR

Alexandre BARACHANT

THÈSE DIRIGÉE PAR **Christian JUTTEN** ET

CODIRIGÉE PAR **Stéphane BONNET** ET **Marco CONGEDO**

PRÉPARÉE AU SEIN DU

Laboratoire Électronique et système pour la santé CEA-LETI
DANS L'École Doctorale : **EEATS**

Commande robuste d'un effecteur par une interface cerveau-machine EEG asynchrone

THÈSE SOUTENUE PUBLIQUEMENT LE **28 Mars 2012**,

DEVANT LE JURY COMPOSÉ DE :

Prof. Jean-philippe LACHAUX

Directeur de Recherche, INSERM Lyon, Président

Prof. Hichem SNOUSSI

Université technologique de Troyes, Rapporteur

Dr. Maureen CLERC

INRIA Sophia Antipolis, Rapportrice

Dr. Antoine SOULOUMIAC

Ingénieur CEA-LIST Saclay, Membre

Prof. Christian JUTTEN

Gipsa-LAB CNRS Grenoble, Membre

Dr. Marco CONGEDO

Gipsa-LAB CNRS Grenoble, Membre

Dr. Stéphane BONNET

Ingénieur CEA-LETI Grenoble, Membre

Prof. Alim-louis BENABID

Conseiller scientifique CEA-LETI Grenoble, Invité



Table des matières

1	Introduction	11
1.1	Contexte	11
1.2	But de la thèse et Approche suivie	14
1.3	Organisation du manuscrit	14
I	État de l’art	17
2	Les interfaces cerveau-machine EEG	19
2.1	Principe	19
2.2	Mesure de l’activité cérébrale	20
2.2.1	Différentes modalités de mesure de l’activité cérébrale	20
2.2.2	Électroencéphalographie	21
2.2.3	Les électrodes	23
2.2.4	Bruit/Artefact	25
2.3	Principaux types d’ICMs EEG	26
2.3.1	Introduction	26
2.3.2	Potentiels Évoqués	27
2.3.3	Imagerie motrice et modulation cérébrale volontaire.	30
2.3.4	ICMs de type ”Wolpaw”	33
2.3.5	ICM Hybride	33
2.4	Stratégies de commande d’effecteurs	33
2.4.1	Commande élémentaire	34
2.4.2	Commande à mémoire temporelle	34
2.4.3	Commande partagée	34
2.4.4	À propos des stratégies de commandes	35
2.5	Couplage ICM et effecteur	35
2.5.1	Fauteuils roulants et robots mobiles	35
2.5.2	Bras robotisés	36
2.5.3	Applications domotiques	36
2.6	Les challenges	36
3	Traitement du Signal pour les ICMs EEG	39
3.1	Introduction	39
3.2	Pré-traitement et extraction de caractéristiques	40
3.2.1	Filtrage fréquentiel	40
3.2.2	Filtrage spatial	40
3.2.3	Filtrage spatial par Common Spatial Pattern	41
3.2.4	CSP multi-classe	45
3.2.5	Filtrage spatio-fréquentiel	46
3.2.6	Extraction de caractéristiques	47
3.3	Classification	49
3.3.1	Introduction	50
3.3.2	Analyse Discriminante Linéaire de Fisher (FLDA)	50
3.3.3	Séparateur à vaste marge (SVM)	53
3.3.4	Régression logistique	54
3.3.5	Validation croisée	55
3.4	Conclusions	56

II Contributions Algorithmiques	59
4 Géométrie Riemannienne : Approche fondée sur la distance	61
4.1 Introduction	61
4.1.1 La chaîne de traitement CSP revisitée	61
4.1.2 Formulation du problème et approche suivie	62
4.1.3 Etat de l'art des approches similaires	62
4.2 La géométrie de l'information	63
4.2.1 Variété des distributions de probabilité	64
4.2.2 Plan tangent, géodésique et distance	64
4.2.3 Métrique	65
4.3 Application aux distributions normales multivariées	66
4.3.1 Métrique	66
4.3.2 Géodésique et distance	67
4.3.3 Moyenne Riemannienne	68
4.4 Classification dans la variété Riemannienne	68
4.4.1 Introduction	68
4.4.2 k plus proches voisins	68
4.4.3 Distance minimale aux centres	70
4.4.4 K -means	72
4.4.5 Résultats expérimentaux	74
5 Géométrie Riemannienne : Espace tangent	79
5.1 Introduction	79
5.1.1 Opérateur exponentiel et logarithmique	79
5.2 Filtrage géodésique	80
5.2.1 Principe	80
5.2.2 Filtrage par projection linéaire	81
5.2.3 Résultats	84
5.3 Classification dans le plan tangent	85
5.3.1 Introduction	85
5.3.2 Sélection de caractéristiques et classification	86
5.3.3 Classification par régression logistique régularisée	87
5.3.4 Kernel-SVM	88
5.3.5 Résultats	88
5.4 Conclusion	90
6 Sélection d'électrodes	93
6.1 État de l'art	93
6.2 Sélection basée sur un critère de détermination	95
6.2.1 Méthode	95
6.2.2 Illustrations	96
6.3 Sélection basée sur un critère de géométrie Riemannienne	98
6.3.1 Méthode	98
6.3.2 Illustration et résultats	101
6.4 Résultats comparatifs	103
6.5 Conclusions	106

III	Contributions Expérimentales	107
7	Mise en place expérimentale d'une ICM	109
7.1	Chaîne expérimentale EEG	109
7.1.1	Matériel	110
7.1.2	Remarques générales	111
7.2	Choix du type d'ICM : le <i>brain-switch</i>	112
7.3	Brain-switch : état de l'art	114
8	Campagne I : Validation de l'apprentissage adaptatif	117
8.1	Paradigme	117
8.2	Chaîne de Traitement	117
8.3	Campagne de test	122
8.3.1	Présentation de l'expérience.	122
8.3.2	Évaluation des résultats.	123
8.3.3	Résultats.	125
8.4	Conclusion	127
9	Campagne II : Couplage avec un effecteur	129
9.1	Paradigme	129
9.2	Chaîne de Traitement	131
9.3	Campagne de Test	135
9.3.1	Présentation de l'expérience	135
9.3.2	Évaluation des performances :	138
9.3.3	Résultats	138
9.4	Conclusion	144
IV	Conclusion	145
V	Annexes	151
A	Estimation des matrices de Covariance	153
A.1	Introduction	153
A.2	Estimation	154
B	Common Spatial Pattern	159
B.1	Approche historique de la CSP	159
B.2	CSP robuste et régularisée	160
C	Séparateur à vaste marge	165
C.1	Données linéairement séparables	165
C.2	Données séparables non linéairement	167
C.3	Données non séparables	168
D	Géométrie de l'information.	171
D.1	Géodésique	171
D.2	Moyenne géométrique	172
D.3	Médiane	174

E	Distances entre matrices de covariance	175
E.1	Distance Euclidienne :	175
E.2	Divergence et distance de Kullback-Leibler	175
E.3	Distance Log-Euclidienne	176
E.4	Récapitulatif	177
F	Liste des publications personnelles	179

Liste des notations

\mathbf{A}	Une matrice.
\mathbf{a}	Un vecteur.
a	Un scalaire.
$ \cdot $	Valeur absolue d'un scalaire.
$ \cdot $	Cardinal d'un ensemble.
$\ \cdot\ _2$	Norme euclidienne d'un vecteur.
$\ \cdot\ _0$	Norme 0 d'un vecteur, nombre d'éléments non nuls du vecteur.
$\ \cdot\ _F$	Norme de Frobenius d'une matrice.
$\langle \cdot \rangle$	Produit scalaire.
$\text{Tr}(\cdot)$	Trace d'une matrice.
$(\cdot)^T$	Transposée d'une matrice.
$P(n)$	Variété des matrices symétriques et définies positives.
$Gl(n)$	Groupe général linéaire, ensemble des matrices $n \times n$ inversibles.
$\mathbf{x}(t)$	Vecteur représentant le signal EEG échantillonné à l'instant t .
\mathbf{X}_i	Matrice représentant une portion de signal EEG (essai ou réalisation).
y_i	Classe de \mathbf{X}_i .
N_t	Nombre d'échantillons temporels d'une réalisation.
C	Nombre d'électrodes.
I	Nombre de réalisations dans une base d'apprentissage.
K	Nombre de classes.
L	Nombre de filtres spatiaux.
\mathbf{v}_i	Vecteur de caractéristiques.
Σ_i	Matrice de covariance de \mathbf{X}_i .
\mathbf{I}_C	Matrice identité de dimension $C \times C$.
$(\cdot)^{(k)}$	Élément relatif à la classe k .

Liste des abréviations

ICM	Interface cerveau-machine.
EEG	Électroencéphalogramme.
IRMf	Imagerie par résonance magnétique fonctionnelle.
SPIR	Spectroscopie proche infrarouge.
MEG	Magnétoencéphalographie.
ECoG	Éléctrocorticographie.
CAR	<i>Common Average Reference.</i>
SSEP	<i>Steady State Evoqued Potential.</i>
SSVEP	<i>Steady State Visual Evoqued Potential.</i>
ERD	<i>Event Related Desynchronisation.</i>
ERS	<i>Event Related Synchronisation.</i>
LED	<i>Light Emitting Diode</i> - Diode électroluminescente.
FLDA	Analyse discriminante de Fisher (<i>Fisher's Linear Discriminant Analysis</i>).
PCA	Analyse en composantes principales (<i>Principal Component Analysis</i>).
ICA	Analyse en composantes indépendantes (<i>Independent Component Analysis</i>).
CSP	<i>Common Spatial Pattern.</i>
CSSP	<i>Common Spatio-Spectral Pattern.</i>
FIR	Réponse impulsionnelle finie (<i>Finite Impulse Response</i>).
IIR	Réponse impulsionnelle infinie (<i>Infinite Impulse Response</i>).
SVM	Séparateur à vaste marge.
CV	Validation croisée (<i>Cross Validation</i>).
SPD	Symétrique et définie positive.
DSP	Densité spectrale de puissance.
k -NN	k plus proches voisins.
MDM	Distance minimale aux centres.
MI	Information mutuelle.
BS	<i>Brain-Switch.</i>
IHM	Interface homme-machine.
VP	Vrai positif.
FP	Faux positif.
FN	Faux négatif.
TPR	<i>True Positive Rate.</i>
PPV	<i>Positive Predictive Value.</i>
RMDM	MDM robuste.
SCM	<i>Sample Covariance Matrix</i>

CHAPITRE 1

Introduction

1.1 Contexte

On nomme Interface cerveau-machine (ICM) tout système permettant d'établir un canal de communication direct entre un cerveau et une machine sans passer par la voie musculaire. C'est à Jacques Vidal que l'on doit le terme d'interface cerveau-machine. Dans l'article [119] "Toward direct brain-computer communication" publié en 1973, il fait état des développements menés à l'Université de Californie et notamment d'un projet pilote : The Brain Computer Interface Project. Ce projet est né de la conviction que le signal enregistré par Électroencéphalogramme (EEG) est constitué en grande partie d'une mixture complexe de signaux d'origine neuronale [119] plutôt que d'une majeure partie de bruit aléatoire comme beaucoup le pensait à l'époque. Ainsi, il apparaît théoriquement possible d'extraire de l'EEG des signatures associées à différents états mentaux, ces signatures pouvant être utilisées pour envoyer des commandes à un ordinateur. A l'époque, le principal frein au développement des interfaces cerveau-machine réside dans la grande quantité de données à traiter en regard de la faible puissance de calcul des ordinateurs disponibles. Ainsi le moindre traitement, aussi basique soit-il, associé à la contrainte temps réel nécessite l'emploi d'un dispositif expérimental conséquent. En 1977, il publie "Real-Time Detection of Brain Events in EEG" [120]. Il s'agit d'un système ICM utilisant les potentiels évoqués visuels (dont l'utilisation n'a que très peu changé aujourd'hui). C'est le premier système ICM fonctionnant en temps réel, qui permet à l'utilisateur de déplacer un curseur dans un labyrinthe virtuel en sélectionnant une des quatre directions élémentaires. Pendant les 15 années qui suivirent la première publication de J. Vidal, les ICMs ne firent l'objet d'aucun développement. C'est à partir des années 90 qu'un regain d'activité se produisit, notamment grâce à J.R. Wolpaw [129]. C'est réellement à partir des années 2000 que le domaine se popularisa. Malgré cet engouement et les progrès réalisés, il n'y a aujourd'hui aucun système d'ICM réellement utilisé en dehors des laboratoires dans lesquels ils sont conçus.

Applications médicales

Le domaine d'application principal des ICMs est le domaine médical. En effet, les ICMs pourraient offrir une nouvelle modalité d'interaction avec le monde extérieur pour des personnes atteintes de graves paralysies ou traumatismes. Certains patients, suite à un traumatisme ou à une maladie dégénérante, se retrouvent sans aucune capacité motrice alors que leur cerveau continue de fonctionner normalement. C'est le cas pour les personnes atteintes du syndrome d'enfermement (*Locked-In Syndrome* ou LIS) : ces personnes se retrouvent entièrement paralysées, sans plus aucune activité musculaire, suite à une lésion du tronc cérébral, causée dans la plupart des cas par un accident vasculaire cérébral (AVC). Leur cerveau étant en parfait état de fonctionnement, les patients atteints de LIS sont parfaitement conscients mais ils sont en revanche incapables de se mouvoir ou de communiquer par la parole. Leur seul moyen de communication se fait à travers les mouvements résiduels des paupières. On compte environ 1,000 personnes atteintes de LIS en France. Pour de tels cas, les ICMs permettraient d'instaurer un nouveau mode de communication ou d'interaction avec l'environnement du patient.

Les ICMs s'inscrivent également dans le champ de la réhabilitation fonctionnelle. Il s'agit alors de palier à une déficience grâce à une prothèse dont les mouvements seraient directement contrôlés par une **modulation volontaire de l'activité cérébrale** [127]. Pour ce type de patient, des solutions adaptées à chaque type de handicap existent déjà et sont bien acceptées. Les ICMs constituent donc une nouvelle modalité qui entre en concurrence avec ces dernières. Les solutions existantes reposent généralement sur l'exploitation de l'activité musculaire résiduelle, en ce sens elles peuvent être très efficaces. Les ICMs devront donc offrir un bénéfice par rapport à ces solutions si on souhaite les utiliser dans ce domaine. A minima, les performances, la robustesse et le nombre de commandes offertes par l'ICM devront égaler celles observées par les autres modalités, ce qui n'est pas encore le cas à l'heure actuelle.

Applications grand public

Pour les applications grand public, le domaine des jeux vidéo est le principal concerné. Les ICMs constituent ainsi une nouvelle modalité d'interaction pour les joueurs. En particulier, les ICMs peuvent offrir une expérience de jeu totalement inédite et à contre courant de la tendance actuelle (détection des mouvements du joueur). En revanche, les conditions requises pour une acceptation large de cette technologie représentent un réel challenge. Ainsi, en plus de la performance et de la robustesse, l'ergonomie et la réactivité, l'ICM devra être irréprochable, sans quoi même le plus enthousiaste des joueurs délaissera cette nouvelle "manette" pour l'ancienne. A l'heure actuelle, seule la société Emotiv¹ propose un casque d'enregistrement de l'activité cérébrale et une suite logicielle destinée à cet effet, mais ne semble pas encore offrir les conditions suffisantes pour convaincre le public ciblé.

Intégrant les limitations des ICMs actuelles, les recherches s'orientent vers de nouvelles applications à destination du grand public, regroupées sous la bannière des ICMs dites "passives". Constituées de l'ensemble des applications impliquant le monitoring de l'état mental de l'utilisateur, ces ICMs ne font pas intervenir le sujet de manière active et relâchent ainsi les contraintes sur la réactivité et les performances. Ces ICMs effectuent le suivi de la charge mentale, du stress, de la somnolence ou de tout autre état général de l'utilisateur et agissent en conséquence, par exemple, sur l'environnement du sujet. Ainsi, il serait possible de mesurer l'état de vigilance d'un conducteur et de l'avertir ou d'adapter le comportement du véhicule pour en tenir compte.

Un domaine multidisciplinaire

Le domaine des ICMs est fondamentalement multidisciplinaire et fait intervenir un vaste champ de compétences.

- **Neurophysiologie** : une bonne compréhension du fonctionnement du cerveau est absolument nécessaire. Les activités cérébrales (ou tâches mentales) que l'utilisateur génère pour interagir avec l'interface activent des zones particulières du cerveau. La compréhension du fonctionnement de ces activations et la connaissance des zones impliquées est alors indispensable et impose des choix au concepteur de l'ICM.
- **Électronique, mécanique et science des matériaux** : les principales technologies de mesure de l'activité cérébrale sont destinées à une utilisation dans un environnement médical. Ces technologies doivent être adaptées aux contraintes définies par le contexte d'utilisation de l'ICM. Cette adaptation pourra se faire à travers la miniaturisation de l'électronique de mesure et le développement de nouvelles électrodes. L'un des challenges actuels réside dans le développement d'un casque de mesure de

1. www.emotiv.com

l'activité cérébrale fiable et ergonomique.

- **Traitement du signal** : l'ICM doit transcrire l'activité cérébrale en une commande destinée à une machine. Cette opération requiert de nombreux traitements sur les données. Les signaux mesurés étant fortement bruités et très sensibles aux perturbations, cette tâche est particulièrement ardue.
- **Interfaces Homme-Machine** : la mise en œuvre d'une ICM fait intervenir l'utilisateur par le biais d'un système d'apprentissage. Ce dernier doit être simple et ergonomique dans le but de maximiser la qualité de cet apprentissage. De la même manière, une ICM doit se révéler simple d'utilisation afin d'éviter de fatiguer et de lasser l'utilisateur, ce qui se traduirait par une baisse des performances du système.

Pour que la réalisation d'une ICM soit un succès il est absolument primordial de considérer la chaîne dans son ensemble. Bien souvent, on observe un cloisonnement entre les différents domaines, où l'équipe chargée du traitement du signal travaille de manière aveugle sur des jeux de données, fournissant ainsi des algorithmes peu adaptés à une mise en situation réelle, et où l'équipe chargée de la réalisation des expérimentations met en œuvre les algorithmes de l'état de l'art, subissant ainsi leurs défauts sans pouvoir y remédier.

Aborder et traiter conjointement toutes les composantes d'une ICM en plaçant l'utilisateur au centre des préoccupations permet alors la mise en place d'un cercle vertueux où les besoins des utilisateurs et les solutions algorithmiques adoptées convergent et débouchent sur la réalisation d'une ICM performante, robuste et ergonomique.

Les challenges

Les travaux réalisés lors de la dernière décennie permettent aujourd'hui d'avoir le recul nécessaire pour appréhender de manière réaliste les capacités des ICMs et pour identifier leurs limitations.

La modalité de **mesure de l'activité cérébrale** utilisée dans la plupart des travaux, y compris pour cette thèse, est l'Électroencéphalogramme (EEG). L'EEG a l'avantage d'être non-invasif et offre une bonne résolution temporelle, ce qui en fait une modalité bien adaptée à la majorité des applications ciblées par les ICMs, en particulier pour les applications à destination du grand public. En revanche, les signaux ainsi obtenus sont fortement bruités et très sensibles aux perturbations, ce qui complexifie grandement leur interprétation. De plus, la quantité d'information exploitable par l'ICM est relativement faible, ce qui limite fortement le nombre de commandes réalisables. Enfin, le matériel nécessaire à l'acquisition de ces signaux est aujourd'hui encore trop encombrant, coûteux et peu confortable.

La **robustesse**, c'est à dire la capacité de l'ICM à garder un comportement cohérent face aux diverses perturbations que subissent les signaux ainsi qu'aux conditions variables dans lesquelles ils sont enregistrés, est également un point bloquant. Il n'y a aujourd'hui que très peu de réalisations pouvant prétendre à un fonctionnement dans des conditions de vie courante.

Enfin, l'**ergonomie** des ICMs est encore loin d'être suffisante. La mise en place du système de mesure et les temps de calibration et d'apprentissage des algorithmes constituent des freins importants à l'utilisation de cette technologie. En particulier, la réalisation d'une ICM permettant une utilisation quotidienne et sur le long terme constitue un défi majeur du domaine.

1.2 But de la thèse et Approche suivie

Cette thèse initie au sein du laboratoire LE2S (Laboratoire Électronique des Systèmes pour la Santé) du CEA-LETI une telle action dans le domaine des ICMs. Ce laboratoire possédant des compétences à la fois en électronique et en traitement du signal, il constitue un environnement propice au développement complet d'une ICM. Nous avons ainsi cherché à aborder de manière globale la réalisation d'une ICM. Ces travaux ont été menés avec comme ligne directrice le développement d'un système d'ICM utilisable dans un contexte de vie courante, le but étant de réaliser une ICM simple d'utilisation, robuste et ergonomique, permettant le contrôle d'un effecteur avec un temps de calibration minimal.

Nous nous sommes orientés vers une ICM **asynchrone**, permettant à l'utilisateur d'envoyer des commandes à tout instant quand il le désire, ce qui offre ainsi une grande liberté d'action et une meilleure ergonomie. Cette ICM sera fondée sur l'**imagerie motrice**, l'envoi des commandes est ainsi effectuée par l'imagination du mouvement d'un membre. Un *brain-switch* ou interrupteur cérébral a été réalisé et permet à l'utilisateur d'envoyer une commande binaire. Bien qu'élémentaire, ce type d'ICM offre de très nombreuses possibilités d'application, et est considéré comme incontournable dans la mise en œuvre d'un système ICM destiné à une utilisation dans la vie courante. En premier lieu, le *brain-switch* permet à l'utilisateur d'envoyer une commande simple pour indiquer au système qu'il souhaite interagir avec lui. Couplé à une ICM plus complexe, le *brain-switch* permet d'augmenter la robustesse de cette dernière en la gardant silencieuse lorsque l'utilisateur ne souhaite pas interagir avec elle. Il pourra également déclencher l'allumage d'un dispositif externe ou encore permettre à l'utilisateur de choisir une action dans un menu. Un tel système se doit d'être le plus fiable possible. Nous avons également souhaité replacer l'utilisateur au centre des développements, en proposant des méthodes adaptatives permettant d'impliquer plus fortement l'utilisateur et de stimuler le processus d'apprentissage.

La réalisation d'une telle ICM implique le développement d'algorithmes robustes et leurs mises en œuvre expérimentales. Les travaux réalisés comportent deux volets, l'un concerne le développement de nouveaux algorithmes, l'autre concerne la réalisation de campagne de tests. Nous avons donc conduit alternativement des phases de mise en œuvre expérimentale qui nous ont permis d'identifier les faiblesses des algorithmes, et des phases de développement algorithmique dans le but de les corriger.

Ce processus nous a permis de traiter et d'apporter des contributions à l'ensemble des blocs constituant une ICM, comme l'illustre la figure 1.1.

1.3 Organisation du manuscrit

État de l'art. La première partie présente l'état de l'art des interfaces cerveau-machine et elle est constituée de 2 chapitres.

Le chapitre 2 passe en revue les différentes approches menant à la réalisation d'une ICM fondée sur l'EEG. Après avoir dressé la liste des principaux types d'ICMs, ce chapitre décrit le couplage avec des effecteurs, et conclura sur l'identification des verrous actuels.

Le chapitre 3 porte sur les méthodes de traitement du signal faisant aujourd'hui office de référence. Ce domaine étant extrêmement vaste, ce chapitre se concentrera sur les méthodes utilisées dans le cadre des paradigmes fondés sur l'imagerie motrice. La chaîne de traitement classique sera présentée, et l'accent sera mis sur les techniques de filtrage spatial et de classification.

Contributions algorithmiques. La deuxième partie présente les principales contributions algorithmiques réalisées pendant cette thèse. Cette partie comporte 3 chapitres. Notre contribution majeure est l'introduction de la géométrie différentielle appliquée aux matrices

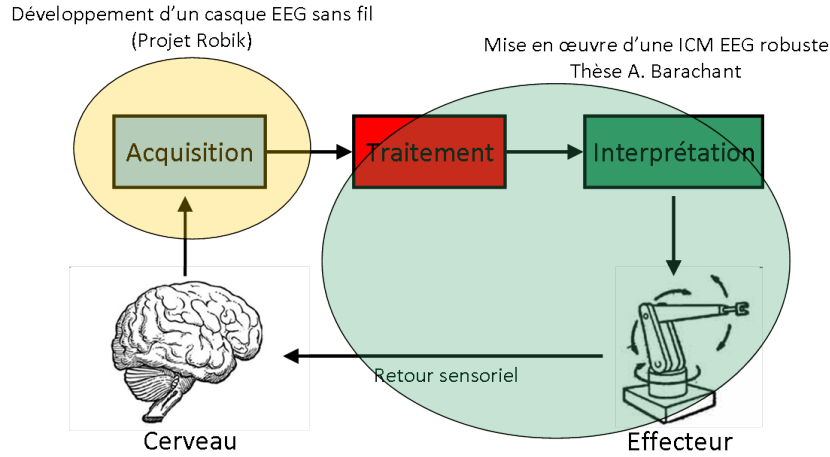


FIGURE 1.1 – Le travail réalisé porte sur la plupart des blocs d'un ICM. Des méthodes de traitement du signal robustes ont été développées, permettant de simplifier grandement la chaîne de traitement. L'interprétation des commandes et le couplage avec un effecteur a été réalisé à travers une application de commande de l'environnement de type domotique. L'interaction avec l'utilisateur et le retour sensoriel a également été travaillé par l'intermédiaire d'une mise en œuvre adaptative afin de stimuler le processus d'apprentissage de l'utilisateur. Le système d'acquisition n'a en revanche pas été abordé durant ces travaux, mais fait l'objet de développements au sein du laboratoire.

de covariance dans le domaine des ICMs, ce qui a permis le développement d'algorithmes originaux impactant l'ensemble de la chaîne de traitement, que ce soit pour l'amélioration des performances, pour plus de robustesse ou d'ergonomie. Cette approche permet une exploitation optimale de l'information spatiale contenue dans les signaux EEG, ainsi la chaîne de traitement a ainsi pu être considérablement simplifiée.

Le chapitre 4 présente les outils de la géométrie différentielle et les applique à la classification de signaux EEG directement à partir d'une mesure de distance Riemannienne entre les matrices de covariance spatiales estimées sur des portions de signaux EEG.

Le chapitre 5 introduit le concept de plan tangent à une variété Riemannienne, permettant ainsi de manipuler les matrices de covariance avec des outils plus évolués et d'obtenir un gain en performance significatif.

Enfin, le chapitre 6 porte sur la problématique de sélection d'électrodes, particulièrement importante pour l'amélioration de l'ergonomie, et propose deux nouvelles méthodes pour répondre à ce besoin, l'une d'elle se fondant également sur les outils de la géométrie Riemannienne.

Contributions expérimentales. La troisième et dernière partie de ce manuscrit présente les 2 campagnes expérimentales menées au cours de ces 3 ans. Cette partie se divise en 3 chapitres.

Le chapitre 7 introduit la chaîne expérimentale mise à disposition par le laboratoire et présente le type d'ICM choisie pour mener à bien ces expériences : le *brain-switch*.

Le chapitre 8 décrit ainsi la première campagne de tests, portant sur la réalisation d'un *brain-switch* avec apprentissage adaptatif, permettant une mise en œuvre simple et rapide de l'ICM.

Enfin, le chapitre 9 présente la deuxième campagne de tests dont l'expérience consistait à coupler le *brain-switch* avec un effecteur afin de permettre à l'utilisateur d'interagir avec son environnement via une application de type domotique.

Première partie

État de l'art

Les interfaces cerveau-machine

EEG

2.1 Principe

Une interface cerveau machine (ICM), ou encore Brain Computer Interface (BCI), est un dispositif permettant d'établir une communication directe entre un utilisateur, ou plus précisément son cerveau, et une machine sans recourir à la voie musculaire. Le principe de fonctionnement est le suivant : à travers la mesure et l'analyse de l'activité cérébrale, on convertit les intentions de l'utilisateur en commandes interprétables par une machine. De par son fonctionnement, le cerveau génère une activité à la fois spécifique en temps, en fréquence et en espace en fonction de l'état mental du sujet. On peut ainsi, en théorie, faire correspondre à chaque état mental une commande destinée à une machine. La détection d'un de ces états mentaux (tâches mentales) entraîne la réalisation, au niveau de l'effecteur, de l'action qui lui est associée arbitrairement.

Il est donc nécessaire, en premier lieu, d'acquérir une **mesure dynamique de l'activité cérébrale**. Il existe plusieurs méthodes pour cela. Dans ce chapitre nous nous limiterons uniquement à la mesure par électroencéphalogramme (EEG) qui est, à plusieurs égards, la technologie la plus employée pour les ICMs. Il s'ensuit une étape de **traitement du signal** ayant pour objectif de **décrypter** les signaux reflétant l'**activité cérébrale** dans le but d'y détecter la réalisation des différentes **tâches mentales**. Cette information est ensuite **traduite en une commande** destinée à un effecteur, qui va ensuite réaliser l'action correspondante en informant l'utilisateur de la bonne prise en compte de la commande via un retour sensoriel. Ce **retour sensoriel** (*feedback*), est principalement visuel [128], parfois auditif [52] et dans de rare cas haptique [25]. La figure 2.1 illustre le principe général d'une boucle ICM.

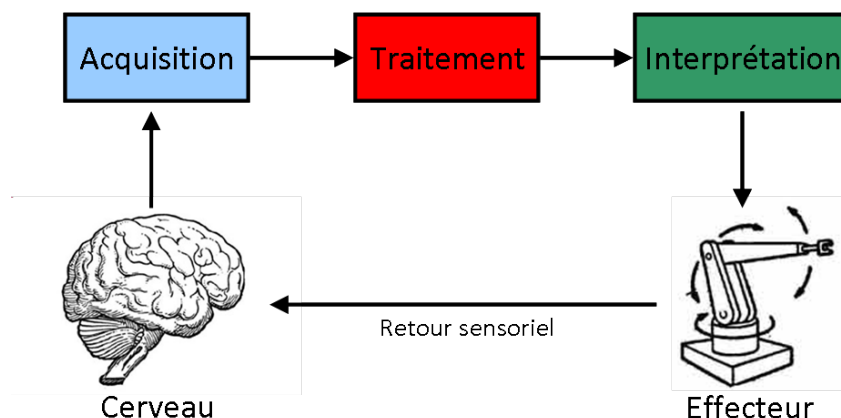


FIGURE 2.1 – Principe de fonctionnement d'une interface cerveau machine. Le cerveau et l'effecteur sont mis en relation à travers une chaîne mettant en œuvre un module d'acquisition de l'activité cérébrale, un module de traitement du signal et un module d'interprétation.

2.2 Mesure de l'activité cérébrale

2.2.1 Différentes modalités de mesure de l'activité cérébrale

Il existe au moins 3 principes, un physiologique et deux physiques, permettant la mesure de l'activité cérébrale.

La réponse hémodynamique est exploitée à la fois par l'imagerie par résonance magnétique fonctionnelle (IRMf) et par la spectroscopie proche infrarouge (SPIR). Lorsqu'une zone du cerveau est active, elle consomme une quantité d'oxygène plus importante, ce qui se traduit par un afflux sanguin dans la zone impliquée. En observant les variations du flux sanguin dans le cerveau, on peut alors en déduire les zones activées. L'IRMf observe cette variation via les propriétés magnétiques de l'hémoglobine contenue dans les globules rouges lors de la désoxygénation [28]. La SPIR observe quant à elle cette variation grâce aux propriétés optiques de l'hémoglobine [122]. Si ces méthodes possèdent des avantages (très bonne résolution spatiale pour l'IRMf, bon rapport signal à bruit pour les deux) les inconvénients sont rédhibitoires pour la plupart des applications d'ICMs. En effet, outre la taille imposante d'un IRM, le principe même sur lequel repose la mesure, à savoir la variation d'oxygénation du sang, intervient avec un délai d'environ 6 secondes après l'activation de la zone [28]. Ce délai est donc trop grand pour envisager une utilisation temps réel. Certaines applications ont cependant vu le jour en utilisant la SPIR [26], mais elles relèvent plus de la preuve de concept que de l'utilisation de cette modalité pour les ICMs.

Les Champs magnétiques : Les neurones communiquent entre eux par l'intermédiaire d'impulsions électriques, ces dernières se traduisant par l'émission d'un champ électromagnétique. Il est alors possible de mesurer une image de l'activité électrique du cerveau soit en mesurant les champs magnétiques qui en émane, soit en mesurant les champs électriques. La technique de mesure des **champs magnétiques** s'appelle la magnétoencéphalographie (MEG). Les champs mesurés étant de l'ordre du femto Tesla (fT), un appareillage conséquent est nécessaire. La mesure doit s'effectuer dans une chambre en mu-métal¹ de manière à supprimer le champ magnétique terrestre. Cependant cette technique de mesure possède une très bonne résolution spatiale (inférieure au millimètre) ainsi qu'une très bonne résolution temporelle, ce qui en fait une méthode de choix pour l'étude du cerveau. On déplore néanmoins un faible nombre d'institutions équipées en MEG, principalement en raison du prix élevé de l'installation. Plus récemment, on a vu apparaître d'autres techniques permettant la mesure d'une activité physiologique via les champs magnétiques [35]. En encapsulant du gel électroactif² dans une électrode que l'on place à proximité du cerveau, on est capable de mesurer un signal reflétant l'activité électrique de ce dernier. Cette technologie est encore loin d'être au point mais à l'avantage d'être miniaturisable et facilement intégrable dans un appareil portable, contrairement à la MEG.

Les Champs électriques : Enfin, la dernière modalité de mesure de l'activité cérébrale consiste à mesurer la résultante des **champs électriques** générés par les neurones, soit de manière invasive en implantant des électrodes directement dans le cerveau ou à la surface du cortex (électrocorticographie, ECoG), soit de manière non-invasive en mesurant les potentiels à la surface du scalp (électroencéphalographie, EEG). Si les techniques invasives offrent les résultats les plus prometteurs [126], ces technologies impliquent de lourdes opérations comportant des risques pour les usagers et pour des raisons éthiques, ne peuvent être utilisées pour certains patients. Cette thèse est tournée vers les applications non-invasives, qui offrent

1. Le mu-métal est un alliage de nickel et de fer qui a une très haute perméabilité magnétique, et constitue donc un excellent matériau pour le blindage magnétique.

2. Gel se déformant sous l'effet des champs électromagnétiques.

certaines des performances limitées mais qui ont pour avantage de faciliter les expérimentations dans un cadre simple à mettre en œuvre et en toute sécurité pour l'utilisateur.

2.2.2 Électroencéphalographie

Généralités

L'électroencéphalographie (EEG) fut inventée en 1924 par Hans Berger. Si le principe est resté globalement inchangé, la technologie a largement évolué, principalement grâce aux avancées en électronique. L'EEG est une simple mesure de différence de potentiel au niveau du scalp. Cette différence de potentiel reflète l'activité électrique locale du cerveau. On la mesure par l'intermédiaire d'électrodes disposées sur le cuir chevelu du patient. On utilise au minimum trois électrodes, une électrode de masse, une électrode de référence et enfin une électrode de mesure. Ces électrodes sont reliées à un amplificateur. Ce dernier va amplifier la différence de potentiel entre l'électrode de mesure et l'électrode de référence, vis-à-vis de la masse commune, prise au niveau de l'électrode de masse.

Le principe d'un amplificateur EEG est relativement simple. Il s'agit d'un amplificateur différentiel, suivi d'un filtrage, d'une numérisation et enfin d'un transfert vers un ordinateur.

Les amplificateurs EEG sont des appareils coûteux, d'une part car ils sont classés en tant que matériel médical et doivent donc satisfaire des normes strictes. D'autre part, même si le schéma de principe est simple, la réalisation de l'appareil est relativement complexe. Les difficultés ne sont pas conceptuelles mais bien technologiques. En effet, les potentiels mesurés sont faibles (de l'ordre de quelques centaines de μV), ce qui nécessite l'emploi de composants bas bruit de bonne qualité. De plus, on utilise, en clinique, un nombre très important d'électrodes (> 128). Ils se posent alors des problèmes de coût, d'intégration, de débit au niveau de la numérisation et du transfert vers un ordinateur.

L'EEG est principalement utilisée en clinique. L'observation de l'EEG permet ainsi de repérer les foyers épileptiques. On l'utilise aussi pour l'observation des cycles du sommeil. De la même manière, cette technique est aussi utilisée pour les patients dans le coma, et entre en compte dans le diagnostic de la mort cérébrale. Plus rarement, on l'utilise en anesthésie pour ajuster le dosage ou surveiller l'état de réveil du patient.

Particularités Fréquentielles

En EEG, la bande passante des signaux est faible (100 Hz environ) et il est rare qu'on retrouve de l'information au delà de 40 Hz. On considère que le signal utile est compris entre 0 et 40 Hz. On distingue principalement 5 bandes fréquentielles (on parle aussi d'ondes) qu'on peut relier à l'état de l'utilisateur ou à la tâche mentale qu'il effectue :

- Les ondes Delta, occupant les fréquences en dessous de 3 Hz, sont généralement associées au sommeil chez l'adulte.
- Les ondes Thêta, entre 4 et 7 Hz, correspondent à un état de détente.
- Les ondes Alpha, entre 7 et 13 Hz, sont les premières ondes remarquables observées grâce à leurs fortes amplitudes. Elles apparaissent de façon très importante lorsque l'on ferme les yeux et que l'on se détend. On les surnomme aussi Mu lorsque ces ondes sont enregistrées au niveau du cortex sensorimoteur : dans ce cas elles apparaissent lors d'un mouvement ou lors de son imagination.
- Les ondes Beta, entre 13 et 30 Hz, sont associées à des périodes d'activité intense du cerveau.
- Les ondes Gamma, entre 26 et 100 Hz, sont pratiquement invisibles en EEG et sont liées à des activités complexes du cerveau. Elles sont encore très peu étudiées.

La position de ces bandes peut varier légèrement, de 2-3 Hz tout au plus, d'une personne à une autre. Les bandes Alpha et Beta sont les bandes de fréquences les plus utilisées en ICM [91].

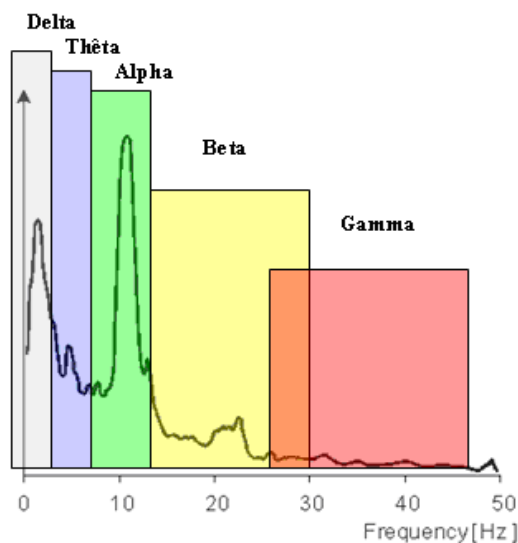


FIGURE 2.2 – Les différentes bandes fréquentielles du signal EEG

Particularités Spatiales

On considère que l'EEG capte la majeure partie des signaux émis par le **cortex cérébral**, le lieu du traitement de l'information. Ce dernier est divisé en zones fonctionnelles, appelées aires, chacune assurant une fonction précise. L'organisation de ces aires est à peu près identique pour l'ensemble des individus, mais elles peuvent toutefois présenter de petites variations. On distingue trois grands types d'aires, représentées figure 2.3 : les aires motrices, les aires sensorielles et les aires d'associations.

Les aires motrices. On distingue deux zones du cortex spécialisées dans la motricité, celles-ci sont situées dans le cortex frontal (en rouge) et sont réparties de part et d'autre du sillon central. La zone principale est l'aire motrice primaire qui occupe toute la partie postérieure du lobe frontal. Une partie précise de cette aire est dédiée à chaque zone du corps et la précision des mouvements est proportionnelle à la surface de la zone en question. La figure 2.4 illustre la répartition des différentes zones dédiées à chaque membre en fonction de leur importance. Ainsi, les zones les plus fortement représentées sont donc celles du visage et de la main. Le rôle de l'aire motrice primaire est d'effectuer les mouvements volontaires.

Les aires sensorielles. Trois aires cérébrales sont spécialisées dans le traitement des données sensorielles : le cortex visuel situé dans le lobe occipital (en vert), le cortex auditif dans le lobe temporal (en jaune) et le cortex somatosensoriel dans le cortex pariétal (en bleu). **Le cortex visuel** est composé d'une part du cortex visuel primaire, qui reçoit les informations en provenance de la rétine et effectue un traitement bas niveau (identification des lignes, des couleurs, ...), et d'autre part du cortex visuel secondaire qui rassemble et associe les informations traitées par le cortex visuel primaire pour obtenir et reconnaître le monde tel qu'on le perçoit. **Le cortex auditif** est organisé de façon similaire au cortex visuel. Il est situé dans le lobe temporal. Il comprend une aire auditive primaire qui identifie les fréquences et une aire auditive secondaire qui reconstitue les sons. **Le cortex somatosensoriel** est organisé de façon similaire au cortex moteur et s'occupe du traitement des informations sensorielles en provenance des membres (on parle souvent de cortex

sensorimoteur pour désigner ces deux entités rassemblées).

Les aires d'association. Les aires d'association représentent la majeure partie du cortex cérébral chez l'homme et sont le principal facteur de la taille de son cerveau. En fait, ce terme désigne toutes les aires qui ne sont ni motrices, ni sensorielles, leurs fonctions sont donc très diverses.

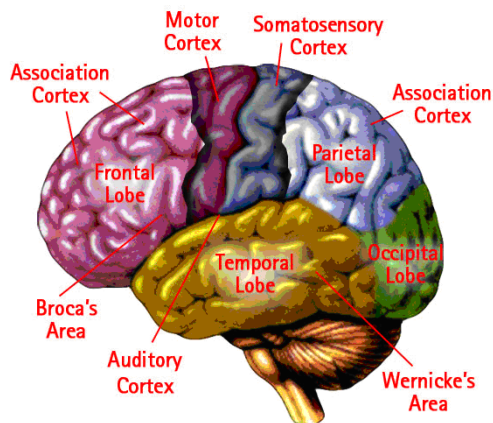


FIGURE 2.3 – Les différentes aires du cerveau

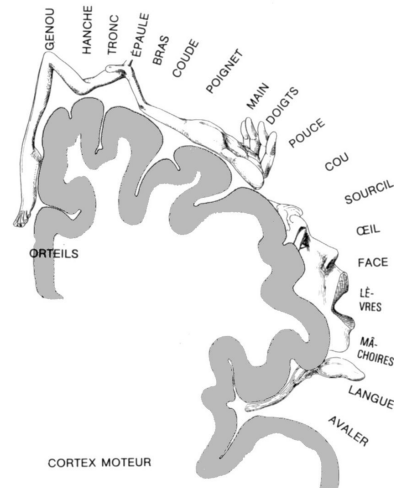


FIGURE 2.4 – Répartition des zones dédiées à chaque membre en fonction de leur importance dans le cortex moteur.

En ICM EEG, on s'intéresse majoritairement aux zones du cortex sensorimoteur [93] (le long de la césure centrale) et également au lobe occipital qui accueille les zones dédiées à la vision (cortex visuel). En effet, de par son organisation somatotopique³, le cortex sensorimoteur permet d'activer volontairement des zones distinctes spatialement. Ainsi, il est plus aisé de produire des tâches mentales discriminables. De son côté, le cortex visuel est le lieu où naissent les potentiels évoqués visuels (cf section 2.3.2). Cette zone est donc impliquée dans toutes les ICMs se fondant sur des stimulations visuelles de l'utilisateur.

2.2.3 Les électrodes

Différents types d'électrodes

Il existe une grande variété d'électrodes utilisées en EEG que l'on emploie selon les contraintes auxquelles on est soumis. Ainsi, pour un usage rapide et de courte durée, on utilise des **électrodes tampon**. Ces électrodes sont constituées d'un petit dôme de 2cm en argent chloruré que l'on recouvre d'un tampon de tissu imbibé avec une solution saline. On pose ensuite ces électrodes sur le cuir chevelu après un rapide dégraissage et l'application d'une pâte conductrice. Elles sont maintenues par un casque en caoutchouc épousant la forme du crâne. La solution saline séchant rapidement, il faut les humidifier fréquemment.

On trouve ensuite les **électrodes cupules**. Ce sont de petits disques d'un demi cm en argent ou en or. La forme en cupule de l'électrode permet de les remplir de pâte conductrice et de les appliquer facilement sur le crâne. Elles sont ensuite collées directement sur le crâne

³. représentation du corps humain au sein d'une structure nerveuse permettant la discrimination spatiale des impressions sensibles

à l'aide de gaze imbibée de colle. Cette technologie permet une mesure plus longue, de l'ordre de la journée.

Pour des enregistrements plus longs, ou lorsque le patient est agité (par exemple lors des crises d'épilepsies) et que les électrodes classiques ne tiennent pas en place, on utilise des **électrodes aiguilles**. Elles ont la forme d'aiguilles hypodermiques et sont placées sous la peau, parallèlement à la surface.

Ces électrodes sont dites **passives**, leur fonction est d'assurer le contact avec le scalp et de conduire les potentiels jusqu'à l'amplificateur EEG. Le problème de ces électrodes réside justement dans le câble qui les relie à l'amplificateur. En effet, les potentiels mesurés étant très faibles, les signaux sont extrêmement sensibles au bruit et aux artefacts dus aux mouvements des câbles.

Pour résoudre ce problème, on utilise de plus en plus souvent des **électrodes actives**. L'idée est de rapprocher au maximum électrode et amplificateur. Pour des raisons pratiques, on ne peut pas réduire la taille des câbles, on va donc chercher à intégrer un premier étage d'amplification au sein même de l'électrode. Ceci a pour effet de réduire les artefacts et le bruit dus au câble, mais aussi de permettre d'augmenter l'impédance acceptable⁴ au niveau de l'interface entre l'électrode et le scalp. Ainsi il n'est plus particulièrement nécessaire de préparer la peau (généralement avec une pâte abrasive) avant la pose de l'électrode, ce qui constitue un gain de temps très important.

L'ensemble de ces électrodes sont dites **humides**, car il est nécessaire d'appliquer une solution ou un gel pour assurer le contact électrique entre l'électrode et la peau. Un nouveau type d'électrode commence à faire son apparition. On parle d'**électrodes sèches** car elles n'ont plus besoin de gel de conduction. Elles ont la forme d'une petite brosse qui va se glisser entre les cheveux et aller au contact de la peau. Cette technologie est encore peu employée mais est très prometteuse car elle est susceptible de réduire très fortement le temps nécessaire à la préparation de l'EEG. Le développement d'électrodes sèches efficaces marquerait l'avènement de l'EEG à destination du grand public, en particulier pour les jeux vidéo.

Pose des électrodes

La pose des électrodes doit être particulièrement soignée. En particulier, le positionnement des électrodes est important pour l'interprétation des résultats et la reproductibilité des expérimentations. Les signaux issus des électrodes sont également fortement corrélés si les électrodes sont trop proches. En dépit de cela l'intensité du signal venant d'une zone spécialisée peut diminuer très rapidement si l'on est placé au mauvais endroit. Il est alors difficile d'utiliser les mêmes paramètres d'une expérience à l'autre. Dans le but de faciliter la pose des électrodes, leurs positions ont été normalisées et suivent le système 10-20. La figure 2.5 représente la position des électrodes dans ce système international.

Le principe est le suivant, de profil (figure 2.5-A) on mesure la distance entre le nasion et l'inion, la première rangée d'électrodes est située à 10% de la distance, puis les suivantes à 20%. On fait de même pour la distance entre le centre des deux oreilles et on obtient la position de chaque électrode. Si on souhaite augmenter le nombre d'électrodes, on intercale les électrodes supplémentaires à équidistance de chacune des électrodes du système 10-20 comme l'illustre la figure 2.5-C.

Les positions de l'électrode de masse et de l'électrode de référence ne sont pas normalisées. Cependant, il est d'usage de placer la référence sur le lobe d'une oreille et la masse proche du front, par exemple en A_{Fz} .

4. En EEG clinique, on considère que l'**impédance mesurée** entre une électrode de mesure (passive) et l'électrode de masse doit être inférieure à 5k Ω pour que les signaux EEG soient de bonne qualité.

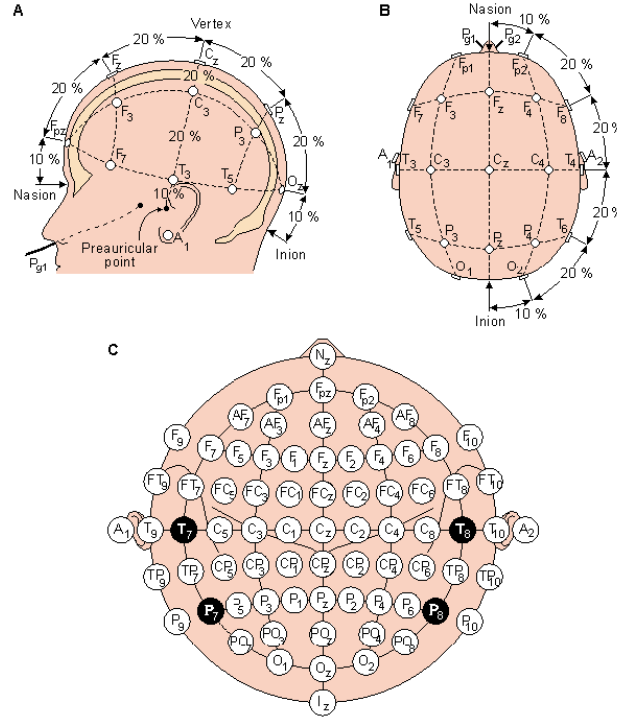


FIGURE 2.5 – Position des électrodes dans le système 10-20

2.2.4 Bruit/Artefact

Du fait même de la mesure, le signal EEG est très sensible à toutes sortes de perturbations étant donné la nature très faible des signaux que l'on cherche à mesurer. On distingue principalement deux bruits de natures différentes.

Tout d'abord le **bruit ambiant** qui émane de l'environnement dans lequel la mesure est effectuée. La principale source de bruit ambiant est le rayonnement électromagnétique du réseau électrique. Cependant la gestion de ce bruit est relativement aisée. D'une part, on considère que le signal cérébral utile que l'on peut mesurer en EEG n'excède pas une fréquence de 40 Hz [92] (sauf cas exceptionnel). Il suffit donc d'appliquer un filtrage passe-bas pour supprimer simplement ce type de bruit. La plupart des systèmes d'enregistrement EEG possèdent un filtrage Notch (coupe bande très spécifique) destiné à cet effet. D'une manière générale, le bruit ambiant n'est pas le bruit le plus gênant en EEG. Qu'il vienne du rayonnement du secteur ou de tout autre appareillage électrique, il réside en général dans des bandes de fréquences plus élevées que celles des signaux EEG. De plus, la source des perturbations est généralement suffisamment éloignée pour que l'ensemble des capteurs soit impacté de façon identique, ce qui rend ces artefacts aisément supprimables, par exemple à l'aide d'un filtrage spatial par *Common Average Reference*⁵ (CAR).

Le deuxième type de bruit est un **bruit de nature physiologique**. Il émane du sujet sur lequel la mesure est effectuée. Parmi ce bruit physiologique, on distingue deux sources, la première est le **cerveau** lui-même ce qui peut paraître paradoxal à première vue. En effet, le cerveau comporte des milliards de neurones qui fonctionnent en permanence. Ceci génère des activités électriques de provenance multiple qui se mélangent et subissent de nombreuses distorsions, notamment lors de la traversée de la boîte crânienne. Les activités spécifiques que l'on cherche à détecter, et qui sont issues d'un groupe réduit de quelques

5. soustraction à chaque signal de la moyenne des autres signaux.

millions de neurones, se retrouvent noyées dans ce bruit provenant de l'ensemble des autres zones du cerveau.

La deuxième source de bruit physiologique est d'origine **musculaire**. En effet, la contraction d'un muscle se trouvant à proximité des électrodes (principalement les muscles du visage) induit l'apparition d'un potentiel qui va se propager et perturber la mesure. L'artefact musculaire rencontré le plus couramment est le battement des paupières. Il induit une forme d'onde caractéristique qui se propage de la zone frontale vers la zone occipitale en s'atténuant. De par ce fait, les signaux issus des différentes électrodes sont perturbés différemment suivant la position de la masse et de la référence. La forme typique d'onde de ce potentiel est donnée figure 2.6.

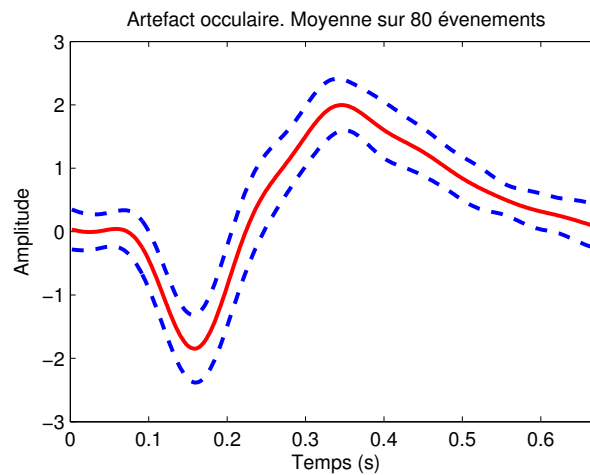


FIGURE 2.6 – Forme typique d'onde d'un artefact oculaire, extrait par PCA et moyenné sur 80 occurrences. L'échelle est normalisée par rapport à la variance du signal EEG. La moyenne est représentée en trait plein, l'écart type en pointillé.

L'artefact oculaire possède un contenu spectral inférieur à 10 Hz. Pour un certain nombre d'ICM, notamment les ICM basés sur l'imagerie motrice, cet artefact est relativement peu gênant puisqu'il empiète peu sur les bandes de fréquences utiles (Alpha et Beta). On se contente alors d'un filtrage passe-haut qui supprime simplement la majeure partie de ces artefacts. Pour les cas où l'on souhaite néanmoins conserver la partie basse des fréquences des signaux EEG, une séparation de sources est possible [60] et s'applique facilement à l'extraction de cette source d'artefact.

L'un des artefacts musculaires des plus gênants est produit lorsque l'on contracte les mâchoires. Cela génère un bruit blanc de variance bien supérieure au signal EEG lui-même comme le témoigne le spectre figure 2.7

Cet artefact est extrêmement dur à supprimer, en revanche il est assez facile à détecter, par exemple sur un critère d'augmentation soudaine de la puissance dans les fréquences élevées (> 50 Hz). Une chaîne de traitement peut être facilement mise en œuvre dans le but de supprimer les portions de signal contenant de tels artefacts.

2.3 Principaux types d'ICMs EEG

2.3.1 Introduction

Toute ICM repose sur un principe commun : **détecter** la réalisation d'une activité cérébrale spécifique à travers le signal EEG [69]. Le cerveau fonctionne suivant des principes

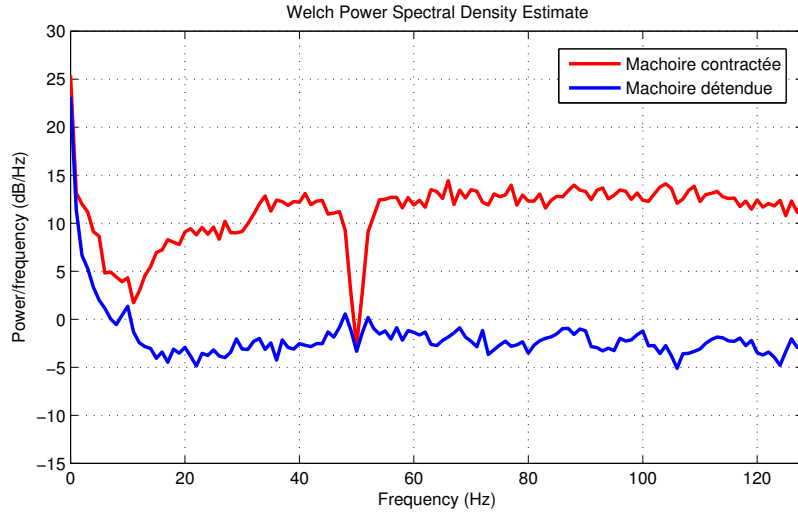


FIGURE 2.7 – Spectre d'un signal EEG lors de la contraction de la mâchoire

différents en fonction des tâches qu'il doit réaliser. On dispose donc d'un panel de tâches mentales [127] induisant la réalisation d'une activité cérébrale spécifique. Ainsi on distingue plusieurs types d'ICMs en fonction des principes physiologiques sous-jacents à la génération des activités spécifiques ainsi qu'au mode de réalisation de l'ICM elle-même.

2.3.2 Potentiels Évoqués

Les potentiels évoqués sont des signaux électriques produits automatiquement par le cerveau en réponse à une stimulation. Dans le cadre des ICMs, cette stimulation est principalement sensorielle. Le plus fréquemment, on stimule visuellement l'utilisateur, par exemple à l'aide d'un flash lumineux [118, 107] et on observe ensuite la réponse générée par le cerveau. Plus rarement, cette stimulation est auditive [42, 66] ou encore somatosensorielle [89]. On parle d'**interface synchrone** pour les ICMs basées sur les potentiels évoqués, car l'analyse des signaux est effectuée en synchronisation avec le stimulus. Pour l'utilisateur, les commandes sont dépendantes de l'arrivée du stimulus, ce qui est parfois frustrant et conduit à des flux d'information assez bas.

Ce type d'ICM ne nécessite pas, ou très peu, d'apprentissage de la part de l'utilisateur. Les réponses utilisées sont induites automatiquement et l'utilisateur demeure peu actif mais doit rester vigilant. L'apprentissage se situe en majorité du côté des algorithmes (cf figure 2.8). Ce type d'ICM peut donc fonctionner avec un niveau de performance optimal très rapidement après sa mise en place, en revanche la marge de progression qui accompagne une utilisation dans la durée reste très faible.



FIGURE 2.8 – Répartition de l'apprentissage entre l'utilisateur et l'algorithmie pour les ICMs de type Potentiel Évoqué. L'apprentissage repose principalement sur les algorithmes.

On distingue deux types de potentiels évoqués : les potentiels évoqués brefs, lorsque la stimulation est parcimonieuse dans le temps avec une réponse cérébrale induite transitoire,

et les potentiels évoqués maintenus (ou Steady State), lorsque la stimulation est continue avec une réponse induite stationnaire.

Potentiels Évoqués Brefs

Ainsi, les potentiels évoqués brefs sont des signaux de durée limitée dans le temps, avec en général une forme d'onde reproductible et bien définie synchronisée sur l'instant de la stimulation. On est donc devant une tâche de détection de la présence d'une activité spécifique vis-à-vis de la non présence de cette activité spécifique, et cela sur un intervalle de temps limité et synchrone avec le stimulus.

Le potentiel évoqué le plus couramment utilisé en ICM EEG est le **P300**. Un onde P300 se produit environ 300 ms à la suite de l'arrivée (cf Figure 2.10) d'un stimulus [101]. Il est localisé au niveau du cortex visuel. Le P300 est utilisé principalement pour des applications de clavier virtuel [107]. En illuminant successivement les différentes lettres d'un clavier virtuel, une réponse P300 est générée lors de l'illumination de la lettre que l'utilisateur veut sélectionner. La détection de ce P300 permet donc de connaître la lettre choisie par l'utilisateur.

Un autre potentiel évoqué utilisé en ICM est le **potentiel d'erreur** [37]. Ce potentiel apparait notamment lorsque l'action effectuée au niveau de la machine est contraire à la volonté de l'utilisateur (par exemple déplacement du curseur à droite alors que l'utilisateur souhaitait le déplacer vers la gauche). Ces potentiels d'erreur pourraient permettre de savoir si l'ICM a mal interprété la volonté de l'utilisateur, et le cas échéant, y remédier [24].

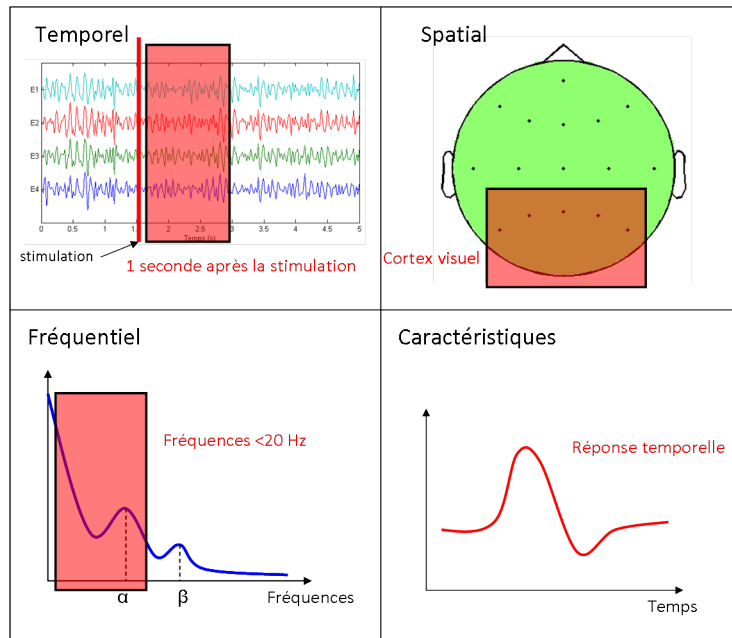


FIGURE 2.9 – Domaine temporel/freq/spatial et caractéristiques utilisées pour la détection du P300. On analyse les signaux sur une fenêtre temporelle après stimulation, sur les électrodes correspondant au cortex visuel et basse fréquence. La réponse attendue est une réponse temporelle typique.

Potentiels Évoqués maintenus

Les Potentiels évoqués maintenus (*Steady State Evoked Potential*, SSEP) sont des potentiels évoqués dont les caractéristiques de la réponse sont constantes dans le temps. Les SSEP sont produits par répétition d'un stimulus sensoriel qui va induire une activation répétée de la zone du cerveau concernée. Il en résulte l'apparition dans les enregistrements EEG d'un signal de fréquence égale à celle de la répétition des stimulus [118].

Lorsque la stimulation est visuelle, ces potentiels sont appelés **SSVEP** (*Steady State Visual Evoked Potential*). On soumet l'utilisateur à une succession de flashes lumineux d'une fréquence donnée et la réponse du cortex visuel se synchronise ainsi sur cette fréquence. Cette réponse est par ailleurs très localisée spatialement. Ainsi on retrouve en O_z la fréquence de stimulation photique de manière très marquée. Dans certaines conditions, on observe une réponse pour des fréquences pouvant aller jusqu'à 90 Hz [51]. L'amplitude de la réponse fréquentielle varie en fonction de l'attention portée au stimulus [63]. Pour utiliser ce phénomène en ICMS, on place ainsi un certain nombre de sources lumineuses (généralement des leds) qui clignotent à des fréquences [118] (ou à des phases [70]) différentes. Si l'utilisateur focalise son regard et/ou son attention sur une de ces sources, la réponse à la fréquence (ou à la phase) correspondante augmente automatiquement. La détection de la réponse prédominante nous indique l'identité de la source lumineuse que l'utilisateur regarde. En associant une action au niveau d'un effecteur à chaque source lumineuse, l'utilisateur peut déclencher la réalisation de l'action simplement en focalisant son attention sur une des sources de stimulation.

Comme pour les potentiels évoqués brefs, il s'agit d'une tâche de détection d'une réponse attendue, mais cette fois non pas dans le domaine temporel mais dans le domaine fréquentiel.

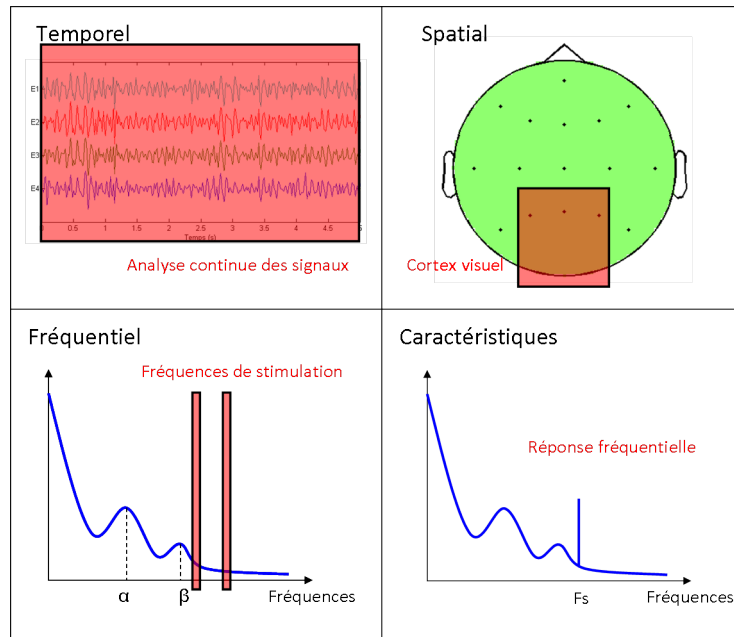


FIGURE 2.10 – Domaine temporel/freq/spatial et caractéristiques utilisées pour la détection des potentiels SSVEP. Analyse continue des signaux, sur les électrodes correspondant au cortex visuel et aux fréquences de stimulation. La réponse attendue est une réponse fréquentielle avec présence d'un pic à la fréquence de stimulation correspondant à la source lumineuse que l'utilisateur regarde.

2.3.3 Imagerie motrice et modulation cérébrale volontaire.

L'imagerie motrice et la modulation volontaire cérébrale constituent un principe de génération d'activité spécifique très utilisé en ICM. Contrairement aux potentiels évoqués, il n'y a pas de stimulation extérieure et c'est l'utilisateur qui, de son propre chef, génère l'activité spécifique. Dans le cas de l'imagerie motrice, on utilise l'imagination du mouvement d'un membre pour activer la zone corticale qui lui est dédiée [93]. Il est également possible d'apprendre à l'utilisateur à contrôler l'activation d'une zone particulière ou d'une bande fréquentielle particulière, on parle alors de modulation cérébrale volontaire.

On appelle **ERD** (*Event Related Desynchronisation*) et **ERS** (*Event Related Synchronisation*) les signaux générés en réponse à ces tâches mentales. L'ERD se traduit par une baisse de la puissance dans une bande fréquentielle et dans une certaine zone, témoins de la désynchronisation d'un ensemble de neurones. L'ERS se traduit par une augmentation de puissance dans une bande fréquentielle et dans une certaine zone, reflétant ainsi une synchronisation du rythme oscillatoire d'un groupe de neurones [92].

Ce sont ces changements de puissance dans une bande spectrale particulière et spatialement localisés que l'on cherche à détecter dans ce type d'ICM. La figure 2.11 montre l'ERD et l'ERS correspondant à un mouvement de la main gauche au niveau de l'électrode C4.

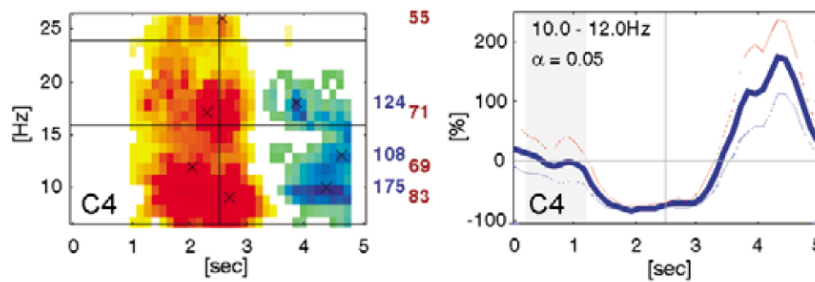


FIGURE 2.11 – Visualisation de l'ERD et de l'ERS pour l'électrode C_4 [49]. Carte temps/fréquence à gauche. Évolution temporelle à droite.

Paradigmes semi-asynchrones

Pour chacune des actions que l'on souhaite réaliser, on associe une tâche mentale d'imagerie motrice, par exemple l'imagination du mouvement de la main droite est souvent associée au déplacement d'un curseur vers la droite. L'expérience se déroule en enchaînant des phases de repos, où l'utilisateur ne peut pas interagir avec l'ICM (le signal n'est pas pris en compte) et des phases d'action (dont le début est marqué par un stimulus), où l'utilisateur doit effectuer la tâche mentale correspondant à la commande qu'il souhaite envoyer à la machine. On parle d'interface semi-asynchrone pour désigner ce type d'interfaces. En effet, contrairement aux ICM basées sur les potentiels évoqués, la tâche mentale n'est pas induite par le stimulus mais est générée volontairement par l'utilisateur, cette dernière est donc **asynchrone** avec la consigne. En revanche, l'utilisateur doit attendre la consigne avant de pouvoir interagir avec la machine.

Un paradigme⁶ couramment utilisé est celui du Graz BCI dont le schéma temporel est donné figure 2.12. les essais de 6 secondes sont séparés par des périodes d'inactivité de quelques secondes. Chaque essai débute par un avertissement sonore suivi de 2 secondes de repos, au terme de ces 2 secondes, on donne la consigne à l'utilisateur pendant 1 seconde.

6. Contexte et ensemble de lois qui régissent le déroulement d'une expérience.

Il dispose ensuite de 3 secondes pour réaliser sa tâche mentale. On parle généralement de paradigme *Cue-based* (sans équivalent en français).

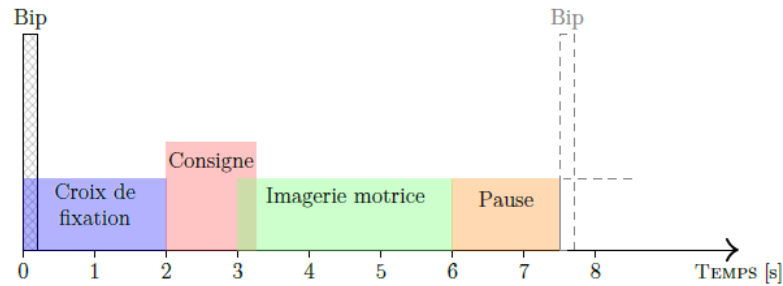


FIGURE 2.12 – Déroulement d'un essai pour un paradigme type du Graz BCI

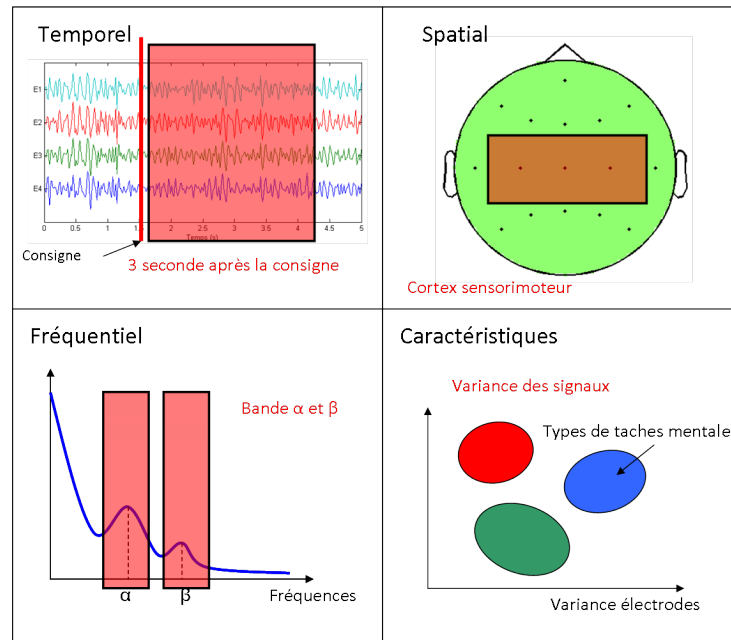


FIGURE 2.13 – Domaine temporel/freq/spatial et caractéristiques utilisées pour la détection pour une ICM de type *Cue-based*. On analyse les signaux sur une fenêtre temporelle après la consigne, sur les électrodes correspondant au cortex sensorimoteur et pour les bandes fréquentielles Alpha et Beta. Les caractéristiques de chaque tâche mentale sont observées à travers la variance des signaux.

Paradigmes asynchrones (*Brain Switch*)

Dans le cas des ICMs asynchrones, l'utilisateur peut interagir avec l'interface en générant à tout moment la tâche mentale souhaitée. On retrouve également ce type d'ICM sous le nom d'ICM *Self-Paced*. Généralement, les ICMs de ce type n'offrent la possibilité d'envoyer qu'une seule commande binaire (commande On/Off), via une seule tâche mentale c'est pourquoi on les appelle **Brain switch** [21].

Ces ICMs représentent le cas typique d'interface destinée à une utilisation continue dans la vie courante. On cherche ainsi à obtenir une commande, certes élémentaire, mais la plus robuste possible. Ce type d'ICM utilise généralement des phénomènes physiologiques bien définis, comme l'augmentation de puissance dans la bande fréquentielle Beta (rebond Beta) suite à un mouvement imaginé du pied [94].

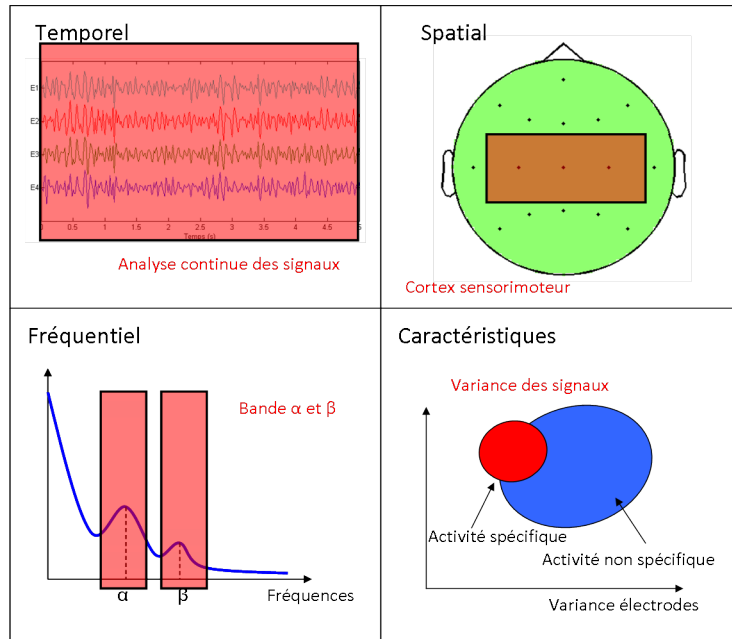


FIGURE 2.14 – Domaine temporel/freq/spatial et utilisées pour la détection pour une ICM de type *Self-paced*. On analyse les signaux en continu, sur les électrodes correspondant aux cortex sensorimoteur et pour les bandes fréquentielles Alpha et Beta. Les caractéristiques de chaque tâche mentale sont observées à travers la variance des signaux.

Remarques sur les ICMs de type *Cue-based* et *Self-paced*. Ces deux types d'ICM font intervenir l'utilisateur de façon plus active que pour les ICMs basées sur les potentiels évoqués. C'est lui qui effectue volontairement les tâches mentales qui lui servent à envoyer des commandes. Ainsi, l'utilisateur doit apprendre à réaliser correctement et de manière reproductible ces tâches mentales. Les algorithmes doivent aussi apprendre à reconnaître ces tâches mentales. L'apprentissage est donc réparti entre l'utilisateur et les algorithmes (cf figure 2.15). On a donc une ICM plus longue à mettre en place mais qui offre une marge de progression relativement importante. Plus elle sera utilisée, plus les performances seront bonnes.

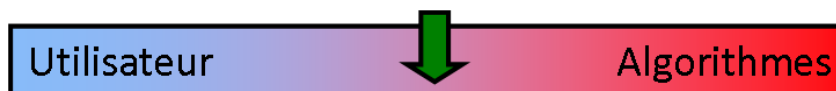


FIGURE 2.15 – Répartition de l'apprentissage entre l'utilisateur et l'algorithmie pour les ICMs de type *Cue-based* et *Self-paced*.

2.3.4 ICMs de type "Wolpaw"

Ce type d'ICM très particulier est développé quasiment exclusivement par l'équipe du professeur J.R. Wolpaw au Wadsworth center. Elles se caractérisent par une exploitation maximale des capacités d'apprentissage de l'utilisateur (cf figure 2.16). Le traitement du signal reste basique mais ce type d'ICM offrent de très belles performances. L'idée principale est d'entraîner l'utilisateur à contrôler chaque degré de liberté d'une ICM à partir de la modulation d'une bande de fréquence au niveau d'un couple d'électrodes. Cet entraînement est facilité par l'emploi d'une régression adaptative. Par exemple, dans le cadre d'un contrôle de curseur en deux dimensions [128], on contrôle le mouvement vertical de manière analogique par :

$$M = a(w_L L_\alpha + w_R R_\alpha + b)$$

où L_α est la puissance de la bande fréquentielle Alpha au niveau de l'électrode C_3 , R_α est la puissance de la bande Alpha au dessus de l'électrode C_4 et a , b , w_L et w_R sont des poids que l'on ajuste par régression. De la même manière, le mouvement horizontal est contrôlé par la bande fréquentielle Beta.

Ce type d'ICM demande un apprentissage très long, allant jusqu'à plusieurs mois pour un contrôle en 3 dimensions [82] mais offre à l'utilisateur une commande analogique directe et naturelle.



FIGURE 2.16 – Répartition de l'apprentissage entre l'utilisateur et l'algorithme pour les ICMs de type "Wolpaw". La majorité de l'apprentissage est effectuée par l'utilisateur qui doit s'entraîner à moduler son activité cérébrale correctement pour contrôler l'ICM.

On notera également que ce principe a été appliqué au contrôle 2D à partir de signaux ECoG [126] et a montré des résultats remarquables pour un temps d'apprentissage réduit à quelques heures.

2.3.5 ICM Hybride

Une **ICM Hybride** est une ICM qui combine plusieurs types d'ICMs dans le but d'augmenter la robustesse de l'interface ou bien sa performance. On couple généralement une ICM synchrone de type potentiel évoqué avec une ICM asynchrone de type Self-paced [90]. On appelle aussi hybride une ICM couplée avec une autre modalité, par exemple à la détection de mouvement ou au suivi oculaire. Un exemple typique d'ICM hybride est donné dans [96]. Un *Brain-Switch* est couplé avec une ICM SSVEP dans le but de contrôler un bras robot. Le *Brain-Switch* apporte la robustesse en permettant d'activer/désactiver le SSVEP, ce dernier apportant un débit de commande plus important. Ainsi l'utilisateur qui souhaite interagir avec le bras active le SSVEP grâce au *Brain-Switch*, fixe la LED désirée pour envoyer la commande puis désactive le SSVEP via le *Brain-Switch*.

2.4 Stratégies de commande d'effecteurs

Après la détection de la réalisation de la tâche mentale, il est nécessaire de transformer cette information en une commande destinée à l'effecteur. Mettre en place une stratégie de commande, c'est définir un cadre qui va décrire la manière dont les ordres envoyés par l'ICM seront interprétés par la machine. Outre la conversion en commande intelligible

par l'effecteur, la stratégie de commande a comme principal but l'augmentation de la robustesse [99] et/ou du nombre de degrés de liberté [46]. Généralement, on peut considérer toute étape de post-traitement, c'est-à-dire après l'étape de classification/détection de la tâche mentale, comme faisant partie d'une stratégie de commande.

Soit c_i la commande destiné à l'effecteur, on distingue principalement trois types de stratégie de commande.

2.4.1 Commande élémentaire

Cette stratégie tient uniquement compte de la sortie y_i de l'algorithme de détection des tâches mentales et établit une relation directe entre cette dernière et la commande c_i . On adapte éventuellement la sortie pour qu'elle soit sous une forme adaptée à la commande de l'effecteur. Soit $g(.)$ une fonction quelconque, la commande s'écrit simplement :

$$c_i = g(y_i)$$

La majorité des systèmes utilise cette stratégie de commande. C'est le cas par exemple pour [81] où la sortie de l'algorithme de détection est directement utilisée pour contrôler le déplacement 2D d'un curseur, ou encore pour [57] où la détection des fréquences SSVEP induit la composition d'un numéro de téléphone. Si cette stratégie est la plus simple, elle montre ses limites rapidement au regard du faible nombre de degrés de liberté atteignable en ICM ainsi que de la faible robustesse de ce dernier.

2.4.2 Commande à mémoire temporelle

Dans ce cas précis, la commande prend en considération les commandes des instants précédents. On peut donc écrire :

$$c_i = g(y_i, \dots, y_{i-n})$$

avec n le facteur de mémoire. Le but peut être l'augmentation du nombre de commandes disponibles à partir d'un nombre limité de tâches mentales, en codant par exemple les commandes en fonction des enchaînements temporels de tâches ou, si une seule tâche mentale est disponible, le temps de maintien de cette tâche mentale. Ainsi l'algorithme de détection reste le même mais la détection d'une des tâches n'induit pas la réalisation de la même action. C'est le cas pour [86] où la commande de fermeture/ouverture d'une main artificielle est codée sur le maintien court d'une unique tâche mentale, alors que la commande de flexion/extension du bras est codée sur un maintien long de la tâche mentale.

2.4.3 Commande partagée

On parle de commande partagée lorsque cette dernière est obtenue à la fois à partir de y_i mais aussi à partir d'un autre jeu de variables, indépendantes de y_i , que l'on notera pour l'occasion \mathbf{q} . On a alors :

$$c_i = g(y_i, \mathbf{q})$$

\mathbf{q} étant un ensemble de variables dépendantes du contexte dans lequel évolue l'ICM.

Par exemple, dans la réalisation d'une chaise roulante à commande partagée, \mathbf{q} peut émaner d'un système de détection d'obstacles [99]. La commande partagée permet dans ce cas d'invalider ou de corriger des commandes envoyées par l'ICM si le système de détection d'obstacles les juge potentiellement dangereuses et donc irréalisables. La commande est donc partagée entre l'ICM et le système de détection d'obstacles. Le but de ce système est d'augmenter la robustesse des commandes.

2.4.4 À propos des stratégies de commandes

Ces différentes stratégies de commandes trouvent leurs origines dans les limitations des ICMs EEG. La commande simple est le choix le plus naturel pour l'utilisateur puisqu'à chaque tâche mentale correspond une action. Avec de l'entraînement, l'utilisateur arrive à associer rapidement l'action à la tâche mentale et cette dernière se réalise alors presque inconsciemment. Cette approche montre ses limitations dès qu'il est question d'utiliser plus de quatre actions différentes et ce, principalement à cause des limitations de l'EEG qui possède une mauvaise résolution spatiale rendant ainsi les tâches mentales difficilement discriminables entre elles.

De plus, certaines de ces stratégies évoluées ne sont pas très ergonomiques. Ainsi, il est déstabilisant pour l'utilisateur qu'une même tâche mentale puisse donner lieu à plusieurs actions différentes. Le processus d'association inconsciente tâche mentale/action est alors difficilement réalisable voire impossible et l'utilisation de l'ICM requiert alors un degré élevé de concentration.

La voie à suivre semble être celle du contrôle partagé tel qu'il peut être décrit dans [99]. On conserve le paradigme "une tâche mentale = une action", tout en déléguant des responsabilités à l'effecteur pour soulager l'utilisateur des tâches complexes et répétitives.

2.5 Couplage ICM et effecteur

Si le but de beaucoup d'ICM est le couplage avec un effecteur dans le monde réel, les premières réalisations sont plutôt récentes (parmi les premières [84], en 2004). On distingue principalement 3 types d'effecteurs utilisés en ICM : les fauteuils roulants et robots mobiles, les bras robotisés et les applications domotiques.

2.5.1 Fauteuils roulants et robots mobiles

Les applications de contrôle de fauteuils roulants et de robots mobiles ont en commun le fait que ces derniers évoluent tous dans un environnement 2D parsemé d'obstacles. Bien que le but à atteindre soit commun, on dénote une grande diversité dans les stratégies pour y parvenir.

Parmi les ICMs synchrones (basées sur les potentiels évoqués), on dénombre surtout des commandes de fauteuils roulants [79, 56, 100, 105, 106]. Dans [105, 106], la commande est de haut niveau : à travers une interface dédiée à un paradigme de type ICM P300, l'utilisateur choisit un lieu parmi une liste. Le fauteuil roulant va alors se rendre dans le lieu sélectionné (cuisine, salon, etc) de manière autonome. Dans [100] la commande est de bas niveau : l'utilisateur choisit simplement (toujours via une ICM de type P300) la direction à prendre parmi une liste de 8 directions. Dans [56], l'interface P300 est fusionnée avec un module de réalité augmentée qui permet de sélectionner directement le lieu que l'on souhaite atteindre dans l'environnement proche.

Parmi les ICMs asynchrones fondées sur l'imagerie motrice, on compte un certain nombre de réalisations concernant des fauteuils roulants [99, 43, 83], mais aussi des commandes de robots mobiles [45, 46]. Parmi l'ensemble des réalisations de commande de fauteuil roulant, l'approche la plus prometteuse semble être la commande partagée [99]. On utilise généralement 3 tâches mentales différentes pour ces ICMs, une pour chaque direction dans laquelle doit évoluer le fauteuil. La commande de robot mobile, quant à elle, est généralement une étape intermédiaire à la réalisation de commande de fauteuil roulant. Elle permet de tester des stratégies de commandes plus complexes [84].

2.5.2 Bras robotisés

On dénombre également des réalisations concernant le contrôle de bras robotisés [77, 117, 36, 87]. Ce type d'effecteur offre un moyen d'interagir directement avec l'environnement proche de l'utilisateur, en revanche la tâche reste plus complexe que la commande de fauteuils roulants car le nombre de degrés de liberté à contrôler est plus important. Il existe principalement deux stratégies pour le contrôle de bras robotisé :

1. La première est dite de **haut niveau** : on ne commande pas directement le bras mais plutôt les actions qu'il doit réaliser. Dans [117], l'utilisateur navigue dans une interface d'ordinateur à l'aide d'une ICM SSVEP, et sélectionne la commande qu'il souhaite voir réaliser parmi une liste de commandes préalablement enregistrés (attraper une bouteille, servir un verre, etc).
2. La deuxième est une commande de **bas niveau** où on contrôle chaque degré de liberté du bras robot. Dans [77], on actionne ainsi un bras robotisé en sélectionnant un à un les degrés de liberté et les déplacements associés parmi une liste sur une IHM à l'aide d'une ICM de type SSVEP.

Les commandes de haut niveau ont l'avantage d'être rapides à exécuter, en revanche l'utilisateur ne contrôle pas réellement le bras ce qui limite les capacités d'interactions avec le monde extérieur à une liste prédéfinie. Les commandes de bas niveau sont très éprouvantes pour l'utilisateur et un temps important est nécessaire pour décomposer et commander un mouvement, même simple.

2.5.3 Applications domotiques

Les applications domotiques sont encore peu fréquentes. En effet, les applications sont en général tournées vers la restauration de la mobilité ou de la communication. Cependant, certaines applications domotiques peuvent apporter une certaine forme d'autonomie (par exemple changer la chaîne de la télévision sans avoir besoin d'appeler une infirmière).

Les applications domotiques se tournent plus volontiers vers des ICMs de type synchrones, car elles permettent un nombre de commandes plus étendu. On note ainsi le contrôle d'une maison intelligente via une ICM de type P300 [54] ou encore un appareil permettant de saisir un numéro de téléphone grâce à un clavier SSVEP [57].

2.6 Les challenges

D'un point de vue performance et nombre de degré de liberté, les ICMs EEG restent plutôt limitées. Cette limitation est principalement due aux spécificités de l'EEG, à savoir une faible quantité d'information utile et mesurable et une très grande sensibilité aux perturbations. Au niveau du traitement du signal, on observe une stagnation des performances maximales. Les limites sont toutefois repoussées régulièrement [82], cependant ces avancées sont souvent portées par un apprentissage plus long de la part de l'utilisateur (jusqu'à plusieurs mois), tirant ainsi partie de la capacité d'adaptation de notre cerveau. En revanche il reste deux points susceptibles d'être largement améliorés et pouvant déboucher sur une acceptation plus grande des ICMs : l'ergonomie et la robustesse.

L'ergonomie est un paramètre important dans tout système ICM qui souhaite se démocratiser. Il est nécessaire d'augmenter le confort du dispositif d'enregistrement de l'activité cérébrale. Les systèmes actuels sont pour la plupart filaires, encombrants et contraignants. Ils restreignent les mouvements et induisent fatigue et lassitude pour l'utilisateur ce qui les rend peu utilisables sur de longues périodes ou au quotidien. On note cependant un effort réel dans le développement de casques de mesure EEG sans fil et ergonomique, étape indispensable si

l'on souhaite développer une ICM à destination d'un public plus large. Le temps nécessaire à la mise en place avant utilisation est aussi un point primordial. La pose des électrodes est en général l'étape la plus consommatrice de temps. Ce temps varie énormément en fonction du nombre d'électrodes à poser, il est donc important de réduire au maximum le nombre d'électrodes sans trop perdre sur la quantité d'information utile que l'on mesure. Le développement d'algorithmes de sélections d'électrodes d'intérêt est ainsi une étape souvent indispensable. Dans la même optique, un système BCI se doit d'être doté d'une étape de calibration la plus courte possible et doit pouvoir s'adapter aux changements en cours d'utilisation (adaptatif). Il est également nécessaire de travailler sur la réduction de la fatigue de l'utilisateur en utilisant des paradigmes et des interfaces adaptées et simples d'utilisation. Enfin, l'utilisation de l'ICM doit être la plus simple possible et ne pas nécessiter l'intervention d'un spécialiste. Ainsi les différents paramètres nécessaires au fonctionnement de l'ICM doivent être réglés automatiquement, ou à défaut, à l'issue d'une procédure de calibration simplifiée et unique.

La robustesse de l'ICM, c'est-à-dire sa capacité à conserver un comportement cohérent même lorsque les signaux sont fortement perturbés, est une qualité indispensable en regard de la forte variabilité de l'EEG. Une plus grande robustesse pourra être introduite par l'utilisation de méthodes de traitement du signal prévues à cet effet. Une stratégie de commande adaptée permet également d'améliorer la robustesse de l'ICM. Par ailleurs, les ICMs hybrides semblent être une voie particulièrement intéressante pour l'augmentation de la robustesse du système.

Traitement du Signal pour les ICMs EEG

Le chapitre précédent a présenté un état de l'art des ICMs en EEG. Ce chapitre abordera les différentes méthodes de traitement du signal nécessaire à la réalisation de ces ICMs. Nous nous restreindrons au cas particulier des ICMs basées sur l'imagerie motrice, présentées paragraphe 2.3.3. Ce chapitre se décompose en deux sections, l'une abordant les algorithmes de pré-traitement comme le filtrage spatial ainsi que l'extraction de caractéristiques, et la seconde portant sur les algorithmes de classification.

3.1 Introduction

Du point de vue du traitement du signal, la réalisation d'une ICM consiste à résoudre un problème de détection d'une ou plusieurs activités mentales à partir des signaux EEG, c'est-à-dire un problème de classification. Cette tâche de détection prend alors la forme mathématique suivante. Soient \mathbf{X}_i une portion de signal EEG et y_i une variable discrète représentant le type d'activité mentale réalisée pendant la portion de signal EEG, on cherche une fonction $f(\cdot)$ telle que :

$$y_i = f(\mathbf{X}_i) \quad (3.1)$$

La principale difficulté dans la réalisation d'une ICM est d'établir cette fonction $f(\cdot)$. L'estimation de f se déroule en deux étapes :

1. **Pré-traitement et extraction de caractéristiques** : à partir des signaux bruts, on applique un certain nombre de pré-traitements et on extrait des caractéristiques dans le but de projeter les signaux dans un espace où les différentes classes pourront être discriminées plus aisément. Ainsi en ICM, on applique généralement un filtrage fréquentiel pour se restreindre aux bandes fréquentielles d'intérêt, puis un filtrage spatial pour faire ressortir l'information en provenance de certaines zones. En imagerie motrice, les caractéristiques qui représentent alors ces signaux sont généralement calculées à partir de la variance de ces signaux pré-traités. Soit $g(\cdot)$ la fonction permettant l'extraction de caractéristique, on a :

$$\mathbf{v}_i = g(\mathbf{X}_i)$$

avec \mathbf{v}_i un vecteur de caractéristiques.

2. **Classification** : on applique ensuite sur ces caractéristiques un algorithme de classification qui affectera à chaque réalisation une classe correspondant à une tâche mentale. Soit $h(\cdot)$ la fonction de décision qui réalise la classification, on a :

$$y_i = h(\mathbf{v}_i) = h \circ g(\mathbf{X}_i)$$

D'une manière générale, il y a un couplage entre ces deux étapes. Plus l'algorithme de classification sera simple, plus l'étape de pré-traitement et d'extraction des caractéristiques

sera complexe, et inversement. Ainsi, si on souhaite utiliser un classifieur linéaire, de type FLDA (analyse discriminante de Fisher) ou SVM linéaire (séparateur à vaste marge), l'étape de pré-traitement devra présenter au classifieur les données dans un espace où elles seront linéairement séparables. De la même manière, si on utilise un classifieur non linéaire comme les réseaux de neurones, les pré-traitements pourront être plus simples car le classifieur sera apte à trouver une fonction complexe permettant de discriminer les différentes classes.

La communauté travaillant sur les ICMS est très active dans le développement et l'application de méthodes d'apprentissage automatique (*machine learning*). Un nombre très important de techniques de classification et d'extraction de caractéristiques a déjà été appliqué au domaine, avec plus ou moins de succès. Ainsi, l'article de synthèse [74] recense pas moins d'une vingtaine d'algorithmes de classification couramment utilisés.

3.2 Pré-traitement et extraction de caractéristiques

Les tâches mentales d'imagerie motrice sont principalement caractérisées en fréquence et dans l'espace. Ainsi l'étape de pré-traitement et d'extraction de caractéristiques s'attachera principalement à effectuer des opérations dans ces deux domaines afin de réduire les signaux EEG à des caractéristiques pertinentes.

3.2.1 Filtrage fréquentiel

Les principales bandes de fréquence impliquées dans les tâches d'imagerie motrice sont la bande Alpha (8-12 Hz) et la bande Beta (18-24 Hz). Il est alors naturel de filtrer par un filtre passe-bande les signaux EEG dans le but de se restreindre à l'une de ces deux bandes d'intérêt ou à une bande plus générale entre 8 et 30 Hz [75, 50]. Ce filtrage passe-bande permet également de s'affranchir d'une part importante des artefacts dûs au clignement des paupières ainsi que du bruit 50 Hz.

Les bandes de fréquence impliquées peuvent varier légèrement en fonction des sujets et [19] propose de sélectionner par sujet la bande de fréquence d'intérêt par l'intermédiaire d'un algorithme d'optimisation.

L'analyse temps/fréquence est également couramment utilisée en EEG pour mettre en évidence et visualiser les ERD/ERS (voir section 2.3.3).

3.2.2 Filtrage spatial

On parle de filtrage spatial pour toute technique utilisant les propriétés spatiales d'un ensemble de signaux dans le but d'en réhausser des caractéristiques. En particulier, on parle de **filtrage spatial linéaire** quand le signal filtré est produit par combinaison linéaire des différents signaux mesurés. On note $z(t)$ la valeur du signal filtré spatialement à un instant t par le filtre $\mathbf{w} = [w_1 \dots w_c \dots w_C]^T \in \mathbb{R}^{C \times 1}$:

$$z(t) = \sum_{c=1}^C w_c x_c(t) = \mathbf{w}^T \mathbf{x}(t) \quad (3.2)$$

avec $x_c(t)$ la valeur de l'échantillon temporel à l'instant t issu de l'électrode $c \in \{1 : C\}$. Le choix de \mathbf{w} peut être défini à l'avance (filtrage de type bipolaire ou laplacien) ou en fonction des propriétés statistiques des signaux. Ainsi on peut calculer les coefficients des filtres par une analyse en composantes principales (PCA) ou indépendantes (ICA), principalement dans le but d'extraire et de supprimer les sources d'artefacts [60]. Dans le cadre des ICMS basées sur l'imagerie motrice, le filtrage spatial le plus couramment utilisé est le filtrage par CSP (*Common Spatial Pattern*) [19]. C'est une méthode qui fait aujourd'hui office de référence.

3.2.3 Filtrage spatial par Common Spatial Pattern

La méthode CSP est construite pour les problématiques à deux classes : la CSP cherche à calculer des filtres spatiaux **maximisant la variance du signal dans une condition tout en minimisant la variance du signal dans l'autre condition**. Historiquement, cette technique de filtrage spatial est dérivée d'une méthode statistique introduite par K. Fukunaga [41] et a été ensuite utilisée pour le traitement des signaux EEG [65]. Elle s'est ensuite largement popularisée dans le domaine des ICMs grâce à H. Ramoser et J. Müller-Gerking [104]. L'approche historique de la CSP est présentée en annexe B.1. Nous avons choisi de présenter la CSP sous la forme d'un problème d'optimisation.

Problème d'optimisation

On note respectivement $\sigma_z^{(1)}$ et $\sigma_z^{(2)}$ l'écart type du signal filtré pour la classe 1 et pour la classe 2. Le critère de la CSP s'exprime alors sous la forme suivante¹ :

$$\max \left(\frac{\sigma_z^{(1)}}{\sigma_z^{(2)}} \right)^2 \quad (3.3)$$

Soit $\Sigma_i \in \mathbb{R}^{C \times C}$ la matrice de covariance estimée sur l'essai \mathbf{X}_i de classe $y_i \in \{1, 2\}$. On calcule tout d'abord les matrices de covariance moyenne² par classe :

$$\Sigma^{(k)} = \frac{1}{|\mathcal{I}^{(k)}|} \sum_{i \in \mathcal{I}^{(k)}} \Sigma_i, \quad k = 1, 2 \quad (3.4)$$

avec $\mathcal{I}^{(k)}$ les indices des essais de classe k . La variance temporelle du signal filtré dans une classe donnée s'obtient d'après équation (3.2) :

$$[\sigma_z^{(k)}]^2 = \mathbf{w}^T \Sigma^{(k)} \mathbf{w} \quad (3.5)$$

A partir des équations (3.3) et (3.5) le critère CSP à maximiser est donné par :

$$J_{CSP}(\mathbf{w}) = \frac{\mathbf{w}^T \Sigma^{(1)} \mathbf{w}}{\mathbf{w}^T \Sigma^{(2)} \mathbf{w}} \quad (3.6)$$

Les solutions maximisant ce critère sont donc données par la résolution du problème suivant :

$$\mathbf{w}_l = \underset{\mathbf{w}}{\operatorname{argmax}} J_{CSP}(\mathbf{w}) = \underset{\mathbf{w}}{\operatorname{argmax}} \frac{\mathbf{w}^T \Sigma^{(1)} \mathbf{w}}{\mathbf{w}^T \Sigma^{(2)} \mathbf{w}} \quad (3.7)$$

On reconnaît alors un quotient de Rayleigh généralisé. L'ensemble de solutions possibles au quotient de Rayleigh ainsi formé satisfait l'équation suivante :

$$\Sigma^{(1)} \mathbf{w}_l = \lambda_l \Sigma^{(2)} \mathbf{w}_l \quad (3.8)$$

avec $\lambda_l = J_{CSP}(\mathbf{w}_l)$ un scalaire et avec \mathbf{w}_l un vecteur propre de la matrice $[\Sigma^{(1)}]^{-1} \Sigma^{(2)}$. On dispose donc d'un total de C solutions, le maximum global étant atteint pour la valeur λ_l maximale, les autres solutions étant des maxima locaux. Étant donné que les \mathbf{w}_l peuvent être définis à une constante multiplicative près, on peut fixer $\mathbf{w}_l^T (\Sigma^{(1)} + \Sigma^{(2)}) \mathbf{w}_l = 1$ pour forcer l'unicité de la solution.

1. On pourra également considérer une minimisation de manière équivalente.

2. Dans le cas présent, c'est la moyenne arithmétique qui est utilisée ; Nous verrons par la suite d'autres manières de calculer la matrice moyenne d'un ensemble de matrices de covariance.

Finalement ce problème peut se reformuler sous la forme d'une diagonalisation jointe non-orthogonale sous contrainte d'unicité :

$$\begin{cases} \mathbf{W}^T \boldsymbol{\Sigma}^{(1)} \mathbf{W} = \mathbf{D}^{(1)} \\ \mathbf{W}^T \boldsymbol{\Sigma}^{(2)} \mathbf{W} = \mathbf{D}^{(2)} \\ s.c. \quad \mathbf{D}^{(1)} + \mathbf{D}^{(2)} = \mathbf{I}_C \end{cases} \quad (3.9)$$

avec $\mathbf{D}^{(k)} = \text{diag}(\{\lambda_l^{(k)}\})$ une matrice diagonale de taille C associée à la classe k , $\lambda_c^{(k)}$ un élément de cette matrice diagonale et $\mathbf{W} = [\mathbf{w}_1 \dots \mathbf{w}_l \dots \mathbf{w}_C]$ la matrice $C \times C$ dont les vecteurs colonnes sont les C filtres spatiaux. On remarquera que $J(\mathbf{w}_l) = \lambda_l = \frac{\lambda_l^{(1)}}{\lambda_l^{(2)}} = \frac{\lambda_l^{(1)}}{1 - \lambda_l^{(1)}}$.

Choix des filtres. On dispose d'une matrice \mathbf{W} composée de C filtres spatiaux \mathbf{w}_l et on cherche généralement à en conserver un nombre réduit $L < C$ (typiquement 6) afin de réduire la dimension du problème. Pour sélectionner les filtres offrant les meilleures performances, on se fondera sur l'observation suivante : la variance du signal filtré spatialement par le filtre \mathbf{w}_l pour la condition (1) a pour valeur moyenne $\lambda_l^{(1)}$. On conservera donc les filtres dont les valeurs propres correspondantes sont maximales (ou minimales). Il est alors d'usage de sélectionner les filtres par paire, chaque paire correspondant aux deux valeurs extrêmes des $\lambda_l^{(1)}$. De manière alternative, on peut trier les filtres en fonction de la valeur $|\lambda_l^{(1)} - 0.5|$, et conserver les L premiers. En effet, un filtre dont la valeur propre est égale à 0.5 pour la classe 1, implique une similarité de variance en moyenne quelle que soit la classe observée ($\lambda_l^{(1)} = \lambda_l^{(2)}$).

Effet des filtres spatiaux sur les signaux. La figure 3.1 illustre l'effet d'un filtrage spatial par CSP sur les signaux EEG. On distingue clairement une augmentation et une diminution de la variance du signal en fonction de la classe.

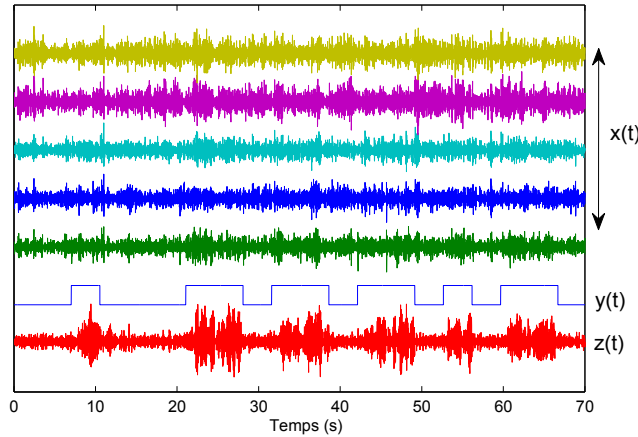


FIGURE 3.1 – Effet du filtrage spatial par CSP sur les signaux EEG. De haut en bas : Les signaux EEG originaux $\mathbf{x}(t)$, la classe des signaux $y(t)$ et un signal filtré $z(t)$.

Effet des filtres spatiaux sur les matrices de covariance intra-classes. Pour un problème à deux dimensions et en observant les ellipses représentant les matrices de covariance moyennes par classe on obtient les résultats présentés figure 3.2. La diagonalisation

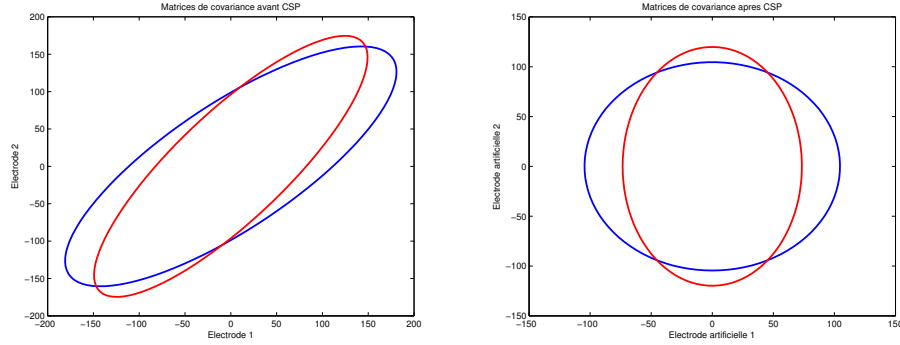


FIGURE 3.2 – Ellipses représentant les matrices de covariance moyenne par classe avant filtrage par CSP (à gauche) et après filtrage par CSP (à droite)

jointe a donc pour effet de rendre orthogonales les composantes principales des signaux de chaque classe. La contrainte d'unicité impose une dispersion faible pour une classe sur un des axes et forte pour l'autre.

Séparation de sources

De manière plus générale, le filtrage spatial par CSP peut également être vu comme une méthode de séparation de sources non-stationnaires. On fait ainsi l'hypothèse que le signal $\mathbf{x}(t)$ est généré par un **mélange instantané de sources indépendantes et non-stationnaires** $\mathbf{s}(t)$ avec la loi de mélange suivante :

$$\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t) \quad (3.10)$$

Les sources $\mathbf{s}(t)$ suivent des lois normales centrées multivariées $\mathcal{N}(0, \mathbf{D}^{(k)})$ avec $\mathbf{D}^{(k)}$ une matrice diagonale différente en fonction des états des sources. Il est montré dans [98] que la séparation peut être obtenue par diagonalisation jointe des matrices de covariance $\mathbf{\Sigma}^{(k)}$ des différents états des sources (après mélange) et le modèle génératif \mathbf{A} peut être estimé au sens du maximum de vraisemblance par $\mathbf{A} = (\mathbf{W}^T)^{-1}$ avec \mathbf{W} un ensemble de filtres spatiaux obtenus par cette diagonalisation jointe.

Dans notre cas, on considère alors des sources non-stationnaires, pouvant prendre deux états différents ($k \in \{1, 2\}$) en fonction du type de tâche mentale réalisée. Alors le filtrage spatial qui permet de remonter aux signaux des sources peut être obtenu par diagonalisation jointe des matrices de covariances spécifiques à chaque classe. On retombe donc sur un problème de diagonalisation jointe identique à celui exprimé dans la CSP. Ce principe a également été utilisé dans [48] pour généraliser la CSP à une problématique multi-classe.

Caractéristiques spatiales

On appelle caractéristiques spatiales (ou spatial pattern [19]) les vecteurs colonnes \mathbf{a}_i de la matrice $\mathbf{A} = (\mathbf{W}^T)^{-1}$. Les coefficients de ces vecteurs représentent les contributions de chacune des sources. L'observation de ces caractéristiques et la comparaison avec la localisation des zones activées par les tâches mentales représente un bon moyen de contrôle sur la qualité finale de la CSP. Par la même occasion, on visualise les coefficients des filtres spatiaux \mathbf{w}_l . La figure 3.3 montre un filtre spatial et une caractéristique spatiale obtenus après application de l'algorithme de CSP.

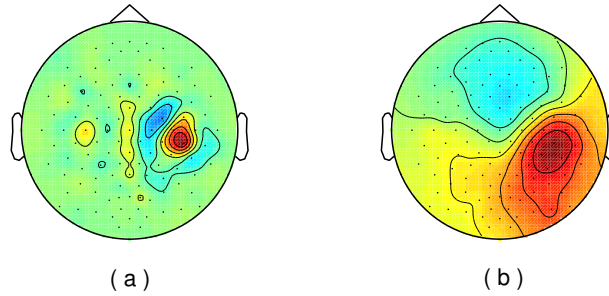


FIGURE 3.3 – Filtre spatial (a) et caractéristique spatiale (b) dans le cas d’un mouvement imaginé de la main gauche. (a) représente la vue topographique des coefficients du filtre \mathbf{w}_l associés à chaque électrode. Il en est de même pour (b) avec les valeurs de la caractéristique spatiale \mathbf{a}_l . Les deux mouvements à discriminer sont un mouvement de la main gauche et un mouvement du pied. Ici on voit clairement que la zone d’intérêt se trouve au niveau de la zone du cortex sensori-moteur réservé à la main, du côté de l’hémisphère droit ce qui est en accord avec le fonctionnement contralatéral du cerveau.

Avantages et inconvénients.

Outre ses performances et sa simplicité, le filtrage spatial par CSP permet de réduire efficacement la dimension de l’espace des caractéristiques. En effet, sa résolution fournit intrinsèquement un critère pour trier les filtres par ordre d’efficacité à partir des valeurs propres. Ce résultat permet donc de faciliter la procédure de sélection de variables d’intérêt, et par la même occasion, de réduire la dimension de l’espace dans lequel on effectue la classification. En revanche, cette méthode possède deux inconvénients majeurs :

Faible robustesse. La technique de filtrage par CSP n’est pas robuste face aux artefacts [75]. Deux causes peuvent être identifiées. La première réside dans l’estimation des matrices de covariance, qui sont estimées généralement par des méthodes sensibles aux artefacts. On utilisera alors des techniques d’estimation robuste des matrices de covariance [130, 2, 124]. La deuxième source de non robustesse est purement mathématique et réside dans la résolution du problème initial de la CSP, donné en équation (3.7). En particulier quand la dimension des matrices est importante, l’inversion et la diagonalisation des matrices de covariance intra-classe peut conduire à des résultats aberrants ou ne correspondant pas avec la réalité spatiale du montage d’électrodes (forte variation de coefficients pour des électrodes proches). Ce type de problème est généralement traité par des méthodes de régularisation [75]. Les différentes techniques employées pour résoudre ces deux problématiques sont détaillées en annexe B.2.

Extension Multi-classe délicate. La CSP est construite à partir d’une problématique à deux classes. En revanche, son extension à plusieurs classes n’est pas nécessairement évidente. Une des solutions est de considérer tous les couples de classes séparément, et de combiner ensuite les caractéristiques ainsi obtenues. Cette solution est peu satisfaisante car elle augmente la dimension de l’espace des caractéristiques et ne considère pas les liens qui lient les classes dans leur ensemble.

3.2.4 CSP multi-classe

Approche directe.

Cette approche généralise directement la CSP au cas multi-classe. On distingue deux méthodes se fondant sur ce principe :

- **1 contre le reste** : On considère une classe par rapport à l'ensemble des autres classes réunies. Si on dispose de $K > 2$ classes, on se rapporte alors à K problèmes de CSP binaire.
- **1 contre 1** : On considère chaque paire de classe individuellement. On a donc $\frac{K(K-1)}{2}$ problèmes de CSP binaire.

On peut alors traiter l'ensemble des problèmes parallèlement et prendre une décision sur les résultats de la classification, par exemple par vote des différents classifieurs. On peut également concaténer les caractéristiques obtenues sur l'ensemble des problèmes binaires, réduire ce nouvel espace par une LDA [30] (cf section 3.3.2) et appliquer une classification multi-classe.

Diagonalisation jointe.

Les solutions précédentes fonctionnent mais ne considèrent pas l'ensemble des classes dans leur globalité. Dans [30], une extension de la CSP binaire a été proposée. De manière analogue au cas binaire, les filtres spatiaux sont obtenus par diagonalisation jointe de l'ensemble des K matrices de covariance intra-classe, ce qui correspond à la résolution d'un problème de séparation de sources non stationnaires [48]. On applique également une contrainte forçant la somme des matrices diagonales à être égale à la matrice identité pour des raisons d'unicité de la solution.

$$\mathbf{W}^T \boldsymbol{\Sigma}^{(k)} \mathbf{W} = \mathbf{D}^{(k)} \text{ s.c. } \sum_{k=1}^K \mathbf{D}^{(k)} = \mathbf{I}_C \quad (3.11)$$

On remarquera que pour $K = 2$, on se ramène bien au cas de la CSP binaire. Lorsque $K > 2$, l'existence d'une matrice \mathbf{W} qui diagonalise parfaitement l'ensemble des matrices n'est plus assurée. On parle alors de diagonalisation jointe approchée car on recherche alors la matrice \mathbf{W} qui diagonalise au mieux les matrices au sens d'un certain critère, par exemple la somme des carrés des termes non-diagonaux. Dans [50], c'est l'algorithme JAD [97] qui est utilisé pour cette diagonalisation jointe approchée.

Sélection des filtres. Contrairement au cas binaire, cette extension ne nous fournit pas de moyen simple de trier les filtres spatiaux [30]. Ce point est pourtant critique, car la majorité des filtres spatiaux est inutile pour la discrimination des différents états mentaux. Il est possible de chercher a posteriori les meilleurs filtres spatiaux après extraction des caractéristiques, par exemple avec un t-test. Bien qu'efficace, cette méthode a un fort coût calculatoire. Dans [30], une méthode de sélection est proposée à partir des valeurs propres calculées. On note $\mathbf{D}^{(k)} = \text{diag}(\{\lambda_l^{(k)}\})$ la matrice obtenue après diagonalisation de $\boldsymbol{\Sigma}^{(k)}$, les valeurs propres sont d'abord normalisées :

$$\tilde{\lambda}_c^{(k)} = \max \left(\lambda_c^{(k)}, \frac{1}{1 + \frac{(K-1)^2 \lambda_c^{(k)}}{1 - \lambda_c^{(k)}}} \right) \quad (3.12)$$

Comme dans le cas de la CSP binaire, ce sont les valeurs extrêmes (grandes ou petites) qui définissent le pouvoir discriminant du filtre. L'heuristique (3.12) découle du fait que deux valeurs différentes peuvent représenter un filtre au même pouvoir discriminant si le

rapport à la moyenne des valeurs propres des autres classes est multiplicativement inverse, c'est-à-dire :

$$\frac{\tilde{\lambda}_c^{(k)}}{\tilde{\lambda}_c^*} = \frac{\lambda_c^*}{\lambda_c^{(k)}} \quad (3.13)$$

avec $\lambda_c^* = \sum_{i \neq k}^K \lambda_c^{(i)} = 1 - \lambda_c^{(k)}$.

Une fois les $\tilde{\lambda}_c^{(k)}$ obtenus, un tri descendant est effectué et on conserve les L premiers éléments de chaque classe. Cette approche, si elle fonctionne relativement bien en pratique, n'est pas fondée théoriquement et effectue une sélection assez approximative des filtres. Pour remédier à ce problème, une seconde approche plus rigoureuse a été développée dans [50]. Elle se fonde sur une estimation de l'information mutuelle entre les signaux filtrés spatialement et leurs classes. Cette estimation est approchée par la formule suivante, sous l'hypothèse que les signaux filtrés soient de variance unitaire et qu'ils suivent des lois gaussiennes :

$$I(y(t), \mathbf{w}^T \mathbf{x}(t)) \approx - \sum_{k=1}^K \log \sqrt{\mathbf{w}^T \boldsymbol{\Sigma}^{(k)} \mathbf{w}} - \frac{3}{16} \left[\frac{1}{K} \sum_{k=1}^K \left(\left(\mathbf{w}^T \boldsymbol{\Sigma}^{(k)} \mathbf{w} \right)^2 - 1 \right) \right]^2 \quad (3.14)$$

On sélectionne ainsi les L filtres qui maximisent l'approximation de l'information mutuelle.

3.2.5 Filtrage spatio-fréquentiel

Le filtrage spatial CSP s'applique à des signaux préalablement filtrés fréquentiellement. Ce sont généralement des filtres passe-bande qui sont appliqués, dont les fréquences de coupure sont soit fixées arbitrairement, par exemple entre 8 et 30 Hz pour englober les bandes Alpha et Beta, soit obtenues par sélection spécifique à chaque utilisateur suivant un algorithme particulier, par exemple une procédure itérative cherchant à maximiser un score [19].

Dans le cadre de l'imagerie motrice, il est connu que les tâches mentales induisent des changements dans l'activité cérébrale à la fois dans le domaine fréquentiel et dans le domaine spatial. Ainsi, l'approche consistant à apprendre ou à optimiser les filtres fréquentiels indépendamment des filtres spatiaux ne semble pas optimale, les deux tâches étant fortement liées.

Dans [71], une telle approche est présentée sous le nom de CSSP (*Common Spatio-Spectral Pattern*). Pour cela, on considérera un filtrage prenant la forme suivante :

$$z(t) = \mathbf{w}_0^T \mathbf{x}(t) + \mathbf{w}_\tau^T \mathbf{x}(t + \tau) \quad (3.15)$$

avec $\tau \in \mathbf{N}^+$. En introduisant une version décalée temporellement du signal, ce filtrage revient à appliquer sur chaque électrode un filtre à réponse FIR de taille $\tau + 1$ avec deux coefficients non-nuls. Le couple de filtres spatiaux $(\mathbf{w}_0, \mathbf{w}_\tau)$ définit alors conjointement le filtrage fréquentiel qui sera appliqué sur chaque électrode.

La résolution de ce problème est identique à la CSP usuelle puisqu'il suffit de concaténer les signaux originaux avec les signaux retardés dans le temps. Soit $\tilde{\mathbf{x}}(t) \in \mathbb{R}^{2C \times 1}$ le signal ainsi obtenu on a :

$$\tilde{\mathbf{x}}(t) = \begin{pmatrix} \mathbf{x}(t) \\ \mathbf{x}(t + \tau) \end{pmatrix} \quad (3.16)$$

En appliquant une CSP sur $\tilde{\mathbf{x}}(t)$, on obtiendra un ensemble de filtres spatiaux $\tilde{\mathbf{W}}$, qu'il suffit de séparer en deux parties pour obtenir les deux filtres donnés en équation (3.15). Soit $\tilde{\mathbf{w}} = [\tilde{w}_1 \dots \tilde{w}_{2C}]^T$ un des filtres spatiaux obtenus par CSP sur $\tilde{\mathbf{x}}(t)$ on aura :

$$\begin{aligned} \mathbf{w}_0 &= [\tilde{w}_1 \dots \tilde{w}_C]^T \\ \mathbf{w}_\tau &= [\tilde{w}_{C+1} \dots \tilde{w}_{2C}]^T \end{aligned}$$

On pourra également expliciter la valeur des coefficients des filtres FIR appliqués à chaque électrode, même si en pratique cela n'est pas nécessaire car ce filtrage est implicitement réalisé en utilisant l'équation (3.15). Soit $\mathbf{h}_c \in \mathbb{R}^{\tau+1}$ la réponse impulsionnelle du filtre pour la c ème électrode, on obtient :

$$\mathbf{h}_c = [\frac{w_{0,c}}{\gamma_c}, 0, \dots, 0, \frac{w_{\tau,c}}{\gamma_c}] \quad (3.17)$$

où $w_{0,c}$ et $w_{\tau,c}$ sont les c èmes coefficients de \mathbf{w}_0 et \mathbf{w}_τ , respectivement, et γ_c est un coefficient tel que :

$$\gamma_c = \frac{\sqrt{w_{0,c}^2 + w_{\tau,c}^2}}{\text{sign}(w_{0,c})} \quad (3.18)$$

Cette approche permet donc d'extraire conjointement des filtres spatiaux et fréquentiels, et permet également de les visualiser de manière à faciliter l'interprétation des résultats. En revanche, on notera l'augmentation de la taille des matrices de covariance à estimer, ce qui pose problème lorsque le nombre d'électrodes est élevé. On a également l'ajout d'un paramètre τ qu'il faut optimiser. Dans [71], cette optimisation est faite à l'aide d'une procédure de validation croisée.

Plus fondamentalement, le filtrage spatial comme le filtrage fréquentiel (par filtre FIR) sont des opérations linéaires sur les signaux. C'est ainsi que la CSSP se ramène aisément à un problème de CSP classique en jouant sur la représentation des données. Partant de ce principe, [33] propose de fusionner à la fois le filtrage fréquentiel, le filtrage spatial et la classification en une seule fonction.

On notera également une autre approche, nommée Filter Bank CSP [3], consistant à découper le signal en plusieurs bandes de fréquences par l'intermédiaire d'un banc de filtres, et à effectuer ensuite un filtrage spatial par CSP sur chacune de ces bandes pour enfin fusionner les caractéristiques fréquentielles ainsi obtenues.

3.2.6 Extraction de caractéristiques

Après un filtrage spatial par CSP (ou une variante), il est généralement d'usage d'utiliser comme caractéristique le logarithme de la variance du signal filtré [19]. Soit $\mathbf{v}_i \in \mathbb{R}^{L \times 1}$ le vecteur de caractéristiques pour l'essai d'indice i , on a :

$$\mathbf{v}_i = \log(\text{var}(\mathbf{z}_i)) = \log(\text{diag}(\mathbf{W}^T \boldsymbol{\Sigma}_i \mathbf{W})) \quad (3.19)$$

avec $\mathbf{W} \in \mathbb{R}^{C \times L}$ une matrice dont les vecteurs colonnes sont les L filtres spatiaux obtenus par CSP et sélectionnés en fonction de leur pouvoir discriminant.

La variance est un choix naturel pour réaliser une classification dans le cadre des ICMs fondées sur l'imagerie motrice. En effet une tâche d'imagerie motrice se caractérise par des changements de puissance dans des bandes fréquentielles particulières. La variance d'un signal représentant également sa puissance, cette dernière est donc apte à caractériser les évolutions induites par les tâches d'imagerie motrice. De plus, le filtrage spatial par CSP optimise également un rapport de variances dans le but de faciliter la discrimination de deux classes.

En considérant que les signaux filtrés spatialement suivent des lois normales (car issus de combinaisons linéaires de vecteurs suivant des lois normales), les variances des signaux filtrés spatialement suivent alors des lois du χ^2 . Or, une grande partie des algorithmes de classification, ou de traitement visant à réduire la dimension de l'espace des caractéristiques, est généralement fondée sur le fait que les données sont distribuées suivant des lois normales. En particulier quand il est nécessaire d'estimer des paramètres statistiques tel que l'espérance ou la covariance des données. La loi du χ^2 n'est pas particulièrement aisée à

manipuler, l'utilisation de la variance n'est donc pas le meilleur choix que l'on puisse faire pour appliquer une classification.

En revanche, il peut être prouvé qu'une loi du χ^2 peut être raisonnablement approximée par une loi log-normale [58]. Ainsi, le logarithme de la variance se rapprochera d'une loi normale et les caractéristiques obtenues pourront être traitées par les algorithmes de l'état de l'art sans que cela ne pose de problèmes.

La figure 3.4 présente la distribution de la variance et de la log-variance d'un signal obtenu par filtrage spatial pour une classe. Pour la variance, une loi log-normale est présentée en superposition. On remarque ainsi que cette distribution est en accord avec la loi log-normale. De la même manière, pour la log-variance, une loi normale est présentée en superposition de la distribution. En dépit des variations observées dans cet exemple, cette approximation est plus juste que celle qui serait faite par l'utilisation de la variance seule (sans passage au logarithme).

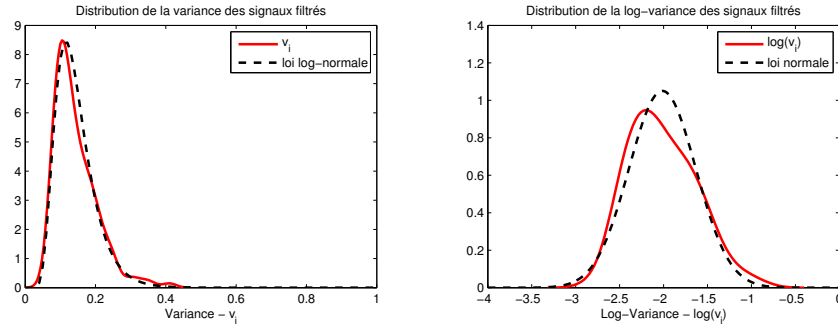


FIGURE 3.4 – Distribution de la variance (à gauche) et de la log-variance (à droite) pour les signaux filtrés par le filtre CSP le plus discriminant.

Lien avec la CSP : Il est intéressant de remarquer que le centre $\mu^{(k)}$ d'une classe dans l'espace des caractéristiques log-variance peut s'exprimer en fonction de la moyenne géométrique des variances des signaux filtrés. Dans ce qui suit, on considèrera, sans perte de généralité, que les signaux sont filtrés par un unique filtre spatial. On omettra ainsi, pour des raisons de lisibilité, l'indice l du filtre utilisé :

$$\mu^{(k)} = \frac{1}{|\mathcal{I}^{(k)}|} \sum_{i \in \mathcal{I}^{(k)}} v_i = \frac{1}{|\mathcal{I}^{(k)}|} \sum_{i \in \mathcal{I}^{(k)}} \log(\sigma_i^2) = \log \left(\sqrt[|\mathcal{I}^{(k)}|]{\prod_{i \in \mathcal{I}^{(k)}} \sigma_i^2} \right) = \log m_g \quad (3.20)$$

avec m_g la moyenne géométrique. De plus, pour une distribution log-normale, la moyenne géométrique est reliée à la moyenne arithmétique par l'équation suivante :

$$m_g = m_a e^{-\frac{1}{2} \sigma_v^2} \quad (3.21)$$

avec m_a la moyenne arithmétique et σ_v^2 la variance des caractéristiques v_i .

Finalement, le centre de la classe k dans l'espace des caractéristiques est relié à la matrice de covariance intra-classe des signaux EEG par :

$$\mu^{(k)} = \log \left(\mathbf{w}^T \Sigma^{(k)} \mathbf{w} e^{-\frac{1}{2} \sigma_v^{(k)2}} \right) \quad (3.22)$$

La distance entre les centres des deux classes s'exprime ainsi par :

$$|\mu^{(1)} - \mu^{(2)}| = \left| \log \left(\frac{\mathbf{w}^T \Sigma^{(1)} \mathbf{w} e^{-\frac{1}{2} \sigma_v^{(1)2}}}{\mathbf{w}^T \Sigma^{(2)} \mathbf{w} e^{-\frac{1}{2} \sigma_v^{(2)2}}} \right) \right| = \left| \log \left(J(\mathbf{w}) e^{-\frac{1}{2} (\sigma_v^{(1)2} - \sigma_v^{(2)2})} \right) \right| \quad (3.23)$$

La distance entre les centres des classes dans l'espace des caractéristiques est donc reliée à la valeur du critère d'équation (3.7) que l'on optimise en CSP. Plus particulièrement, la valeur propre associée à chaque filtre étant égale à $J(\mathbf{w})$, on aura :

$$|\mu^{(1)} - \mu^{(2)}| = \left| \log(\lambda) - \frac{1}{2} \left([\sigma_v^{(1)}]^2 - [\sigma_v^{(2)}]^2 \right) \right| \quad (3.24)$$

En supposant que les variances des deux classes sont sensiblement égales, on pourra alors voir que le logarithme de la valeur propre de chaque filtre quantifie la distance entre les centres des deux classes (pour un filtre donné) :

$$|\mu^{(1)} - \mu^{(2)}| \approx |\log(\lambda)| \quad (3.25)$$

Plus la distance entre les centres des deux classes est grande (si toutefois la variance est constante), plus il sera aisé de les discriminer. Ainsi la valeur propre associée à chaque filtre permet bien de quantifier le pouvoir discriminatif des filtres. De plus, on remarque également que les valeurs faibles ou fortes de λ maximisent cette distance, d'où le choix de sélectionner les filtres dont les valeurs propres sont proches de zéro ou très grandes.

La figure 3.5 présente les distributions pour les deux classes après filtrage spatial par le filtre le plus discriminant. Les traits pointillés représentent l'estimation des moyennes des caractéristiques de chaque classe. Les distributions n'étant pas parfaitement gaussiennes, l'estimation des moyennes ne coïncide pas avec le maximum de la distribution. Dans cet exemple, la distance entre les deux centres $|\mu^{(1)} - \mu^{(2)}| = 1.75$ et le logarithme de la valeur propre associée au filtre $\log(\lambda) = 1.78$. On remarque que l'approximation donnée équation (3.25) se vérifie plutôt bien.

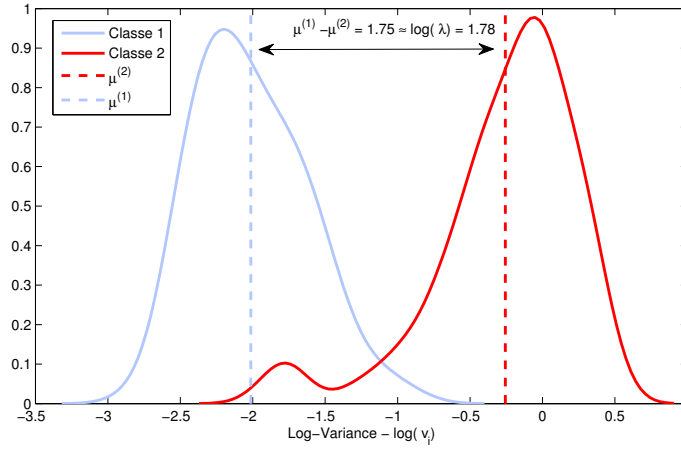


FIGURE 3.5 – Distribution de la log-variance pour les deux classes. La distance entre les moyennes est approximativement égale au logarithme de la valeur propre associée au filtre spatial appliqué sur les signaux.

Utilisé dans ce contexte (caractéristiques de type log-variance) le filtrage spatial par CSP cherche donc à optimiser la distance entre les moyennes de chaque classe, ce qui est sous-optimal car la dispersion des classes n'est pas prise en compte.

3.3 Classification

La section précédente a présenté des techniques de pré-traitement et d'extraction de caractéristiques. Ces caractéristiques représentent de manière pertinente les signaux EEG. A

partir de ces caractéristiques, une règle de décision sera estimée et permettra de classer les signaux EEG en fonction de leur appartenance à une classe e.g. la tâche mentale d'imagination motrice. Nous verrons dans cette section les principales approches utilisées en ICM pour réaliser cette classification.

3.3.1 Introduction

Principe : A partir des caractéristiques extraites à l'étape précédente on souhaite effectuer **une classification**, c'est-à-dire estimer une fonction qui pourra déterminer l'appartenance d'une réalisation à une classe particulière. Soit $h(\cdot)$ une telle fonction, \mathbf{v}_i un vecteur de caractéristiques d'indice i et y_i sa classe, on aura :

$$\hat{y}_i = h(\mathbf{v}_i) \quad (3.26)$$

où \hat{y}_i est l'estimation de la classe de \mathbf{v}_i .

Apprentissage et classification : Afin d'estimer la fonction $h(\cdot)$ on dispose généralement d'une base d'apprentissage, c'est-à-dire d'un ensemble de vecteurs de caractéristiques de chaque classe. Lorsque la classe associée à chacune de ces réalisations de la base d'apprentissage est connue, on dit que la classification est **supervisée**. Si on ne dispose pas de ces informations, la classification est dite **non-supervisée**. Dans ce cas, il est nécessaire d'effectuer une étape de partitionnement qui affectera à chaque réalisation une classe en fonction de la manière dont les données sont réparties dans l'espace des caractéristiques. La classification non-supervisée est rarement utilisée pour les ICMs car elle ne permet pas d'atteindre des performances satisfaisantes.

Le processus de classification supervisée comporte deux étapes.

1. **Apprentissage :** A partir d'une base d'apprentissage, c'est-à-dire d'un ensemble de couples réalisation/classe $\{(\mathbf{v}_1, y_1), \dots, (\mathbf{v}_i, y_i), \dots, (\mathbf{v}_I, y_I)\}$ on va chercher à estimer la fonction $h(\cdot)$ qui permettra d'associer au mieux chacune des réalisations à sa classe respective. Cet apprentissage est généralement guidé par la minimisation d'un critère $\epsilon(\hat{y}_i, y_i)$ témoignant de la justesse de la classe estimée. Ainsi, il est courant de chercher à minimiser l'erreur au sens des moindres carrés :

$$h = \underset{h}{\operatorname{argmin}} \sum_i [y_i - h(\mathbf{v}_i)]^2$$

2. **Classification :** Une fois $h(\cdot)$ estimée, on applique l'équation (3.26) à chaque réalisation de classe inconnue pour en déterminer la classe.

Ce processus est schématisé par la Figure 3.6

Sur-apprentissage : L'utilisation de la classification supervisée expose au risque de sur-apprentissage, c'est-à-dire qu'il est possible que la fonction de classification obtenue à partir de la base d'apprentissage ne soit pas généralisable à des données n'appartenant pas à cette base d'apprentissage. C'est généralement le cas lorsque la dimension de l'espace des caractéristiques où le nombre de paramètres du classifieur sont proches ou supérieurs au nombre de réalisations disponibles dans la base d'apprentissage. On utilise généralement une procédure de **validation croisée** (cf. section 3.3.5) pour contrôler si le classifieur obtenu après apprentissage est généralisable.

3.3.2 Analyse Discriminante Linéaire de Fisher (FLDA)

La FLDA est une méthode d'analyse discriminante qui permet avant tout de trouver des composantes qui maximisent la séparabilité des classes au sens du critère de Fisher. La

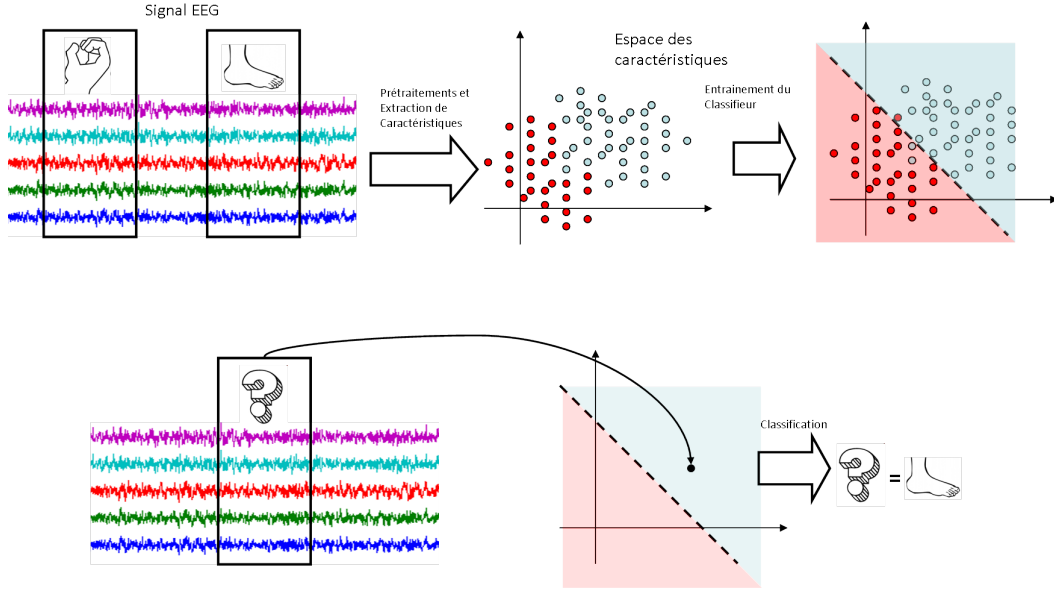


FIGURE 3.6 – Processus de classification. A partir des signaux EEG bruts, on extrait des caractéristiques qui représenteront les signaux de manière pertinente. **Une fonction de décision** $h(\cdot)$ est estimée à partir de ces caractéristiques et permet de découvrir la classe d'une portion de signal de classe inconnue.

FLDA permet de réduire l'espace des caractéristiques par projection sur ces composantes et ainsi conserver uniquement l'information utile à la séparation des classes. Suite à cette projection, il est nécessaire d'appliquer un classifieur afin de prendre la décision sur la réalisation que l'on souhaite classer.

Calcul du projecteur (cas binaire) : Le calcul du projecteur se fait suivant le critère de Fisher, qui cherche à maximiser la distance entre les centres des 2 classes tout en minimisant la dispersion des données après projection. La figure 3.7 illustre ce critère.

Soient $\mu^{(1)}$ et $\mu^{(2)}$ les centres des classes après projection, soient $\sigma^{(1)}$ et $\sigma^{(2)}$ les écarts type des classes après projection, le critère de Fischer s'exprime sous la forme suivante :

$$J_F = \frac{(\mu^{(1)} - \mu^{(2)})^2}{\sigma^{(1)2} + \sigma^{(2)2}} \quad (3.27)$$

Ce critère peut alors être explicité en fonction du projecteur \mathbf{w} :

$$J_F(\mathbf{w}) = \frac{\mathbf{w}(\mu_v^{(1)} - \mu_v^{(2)})(\mu_v^{(1)} - \mu_v^{(2)})^T \mathbf{w}^T}{\mathbf{w}(\Sigma_v^{(1)} + \Sigma_v^{(2)})\mathbf{w}^T} \quad (3.28)$$

où $\mu_v^{(k)}$ et $\Sigma_v^{(k)}$ sont le vecteur moyenne et la matrice de covariance des caractéristiques pour la classe k . On introduit alors la matrice \mathbf{S}_b , nommée *between class scatter matrix*, tel que :

$$\mathbf{S}_b = (\mu_v^{(1)} - \mu_v^{(2)})(\mu_v^{(1)} - \mu_v^{(2)})^T \quad (3.29)$$

et la matrice \mathbf{S}_w , nommée *within class scatter matrix*, tel que :

$$\mathbf{S}_w = \Sigma_v^{(1)} + \Sigma_v^{(2)} \quad (3.30)$$

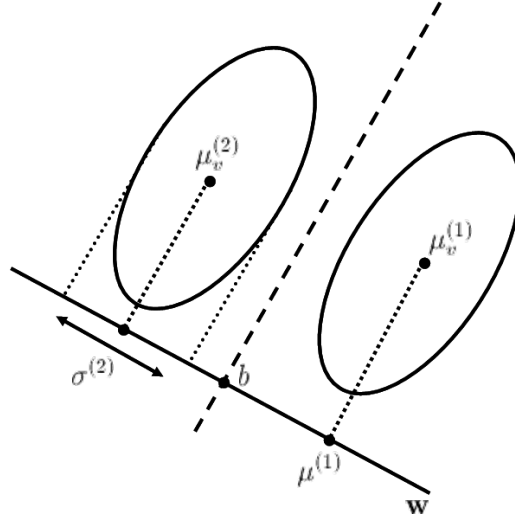


FIGURE 3.7 – Classification par LDA. Les deux classes sont représentées par les ellipsoïdes. On cherche le projecteur \mathbf{w} qui maximise l'écart entre les centres tout en minimisant la dispersion.

Alors le critère³ d'optimisation guidant le calcul du projecteur prend la forme suivante :

$$J_F(\mathbf{w}) = \frac{\mathbf{w} \mathbf{S}_b \mathbf{w}^T}{\mathbf{w} \mathbf{S}_w \mathbf{w}^T} \quad (3.31)$$

On retrouve ici un quotient de Rayleigh généralisé, dont la solution est obtenue par diagonalisation jointe de \mathbf{S}_b et \mathbf{S}_w . La matrice \mathbf{S}_b étant de rang 1, le projecteur est obtenu simplement par l'équation suivante (à un facteur multiplicatif près) :

$$\mathbf{w} = \mathbf{S}_w^{-1} (\mu_v^{(1)} - \mu_v^{(2)}) \quad (3.32)$$

Calcul des projecteurs (cas multi-classe) : Il existe également une extension multi-classe la FLDA. Ainsi on définit la *matrice de covariance inter-classe* \mathbf{S}_b par :

$$\mathbf{S}_b = \frac{1}{K} \sum_{k=1}^K (\mu^{(k)} - \bar{\mu}) (\mu^{(k)} - \bar{\mu})^T \quad (3.33)$$

où K est le nombre de classes et $\bar{\mu}$ est la moyenne de l'ensemble des $\mu^{(k)}$.

La matrice de covariance intra-classe \mathbf{S}_w est définie, par analogie au cas binaire, par la somme de l'ensemble des matrices de covariance moyennes par classe $\Sigma_v^{(k)}$:

$$\mathbf{S}_w = \sum_{k=1}^K \Sigma_v^{(k)} \quad (3.34)$$

Les projecteurs sont également calculés par diagonalisation conjointe de \mathbf{S}_b et \mathbf{S}_w [31]. En particulier, ils sont égaux aux vecteurs propres de $\mathbf{S}_w^{-1} \mathbf{S}_b$. \mathbf{S}_b étant de rang $K - 1$, on utilisera un maximum de $K - 1$ projecteurs pour la réduction de l'espace des caractéristiques.

3. Ce critère est nommé critère de Fisher.

Classification (cas binaire) : Pour le cas binaire, on dispose d'un seul projecteur, il suffit de déterminer un terme de biais afin d'effectuer la classification. N'ayant pas d'a priori sur les données, ce terme de biais est calculé pour être à la moyenne de la distance entre les deux centres des classes :

$$b = \mathbf{w}^T \left(\frac{\mu_v^{(1)} + \mu_v^{(2)}}{2} \right) \quad (3.35)$$

Finalement, la classification s'effectue par la fonction suivante :

$$\hat{y}_i = h(\mathbf{v}_i) = \text{sign}(\mathbf{w}^T \mathbf{v}_i - b) \quad (3.36)$$

On pourra noter que sous certaines conditions, la classification par LDA est optimale dans le sens où elle minimise l'erreur de classification, et plus particulièrement, le classifieur obtenu correspond au **classifieur optimal de Bayes** [31]. Soient les trois conditions suivantes [18] :

1. **Les caractéristiques des classes suivent des lois gaussiennes multivariées.** Cette hypothèse est relativement bien vérifiée dans le cadre de l'imagerie motrice.
2. **Les matrices de covariance de chaque classe sont égales.** Cette hypothèse permet d'affirmer que les classes seront linéairement séparables (sous condition que la distance entre les centres soit suffisamment grande). En pratique, cette hypothèse est rarement vérifiée.
3. **On peut estimer correctement les paramètres des distributions de chaque classe.** Dans le cadre des ICMs, où on dispose de relativement peu de réalisations de chaque tâche mentale, l'estimation des paramètres des distributions est alors une problématique réelle. En revanche, contrairement aux deux hypothèses précédentes, il est possible d'agir sur ce point en utilisant des estimateurs robustes ou en appliquant une régularisation sur la matrice de covariance [18].

Classification (cas multi-classe) : Contrairement au cas binaire, la projection ne permet pas de classer les données. On appliquera alors un classifieur sur les données projetées, par exemple un classifieur naïf de Bayes.

La FLDA est très populaire dans le domaine des ICMs, et elle est généralement utilisée en combinaison du filtrage spatial par CSP [19]. En réalité la classification par FLDA est un choix naturel lorsque que l'on utilise la log-variance des signaux filtrés par CSP comme caractéristique pour la classification. Le critère de calcul de la CSP (3.3) revient à optimiser la distance entre les centres des classes dans l'espace des caractéristiques, cf. équation (3.25).

Le filtrage par CSP construit donc un espace des caractéristiques quasiment optimal pour la FLDA puisqu'il maximise la distance entre les centres des classes. En revanche il ne prend pas en compte la dispersion des classes. Ainsi une amélioration du filtrage spatial est proposée dans [123] afin d'incorporer le critère de Fisher dans la résolution de la CSP.

La FLDA est également très simple à calculer, ce qui en fait une méthode très efficace en terme de coût de calcul. De plus, sa résolution suivant l'équation (3.32) fait intervenir la matrice \mathbf{S}_w qui est **indépendante des classes** (la matrice de covariance de l'ensemble des données est souvent utilisée), il est alors possible de mettre à jour le classifieur de manière non supervisée. Même si cette mise à jour n'est pas optimale, elle permet de prendre en compte les changements sur la distribution des données qui peuvent apparaître au cours d'une session (ou entre deux sessions) [121].

3.3.3 Séparateur à vaste marge (SVM)

Avec ces hypothèses fortes sur les distributions des données, le projecteur obtenu par FLDA est souvent considéré comme non optimal car ne se conformant pas à la réalité

des données. Parmi l'ensemble des classifieurs linéaires, le SVM est généralement considéré comme une des méthodes de classification les plus efficaces.

La problématique que cherche à résoudre le SVM est la suivante : lorsque deux classes sont parfaitement séparables, il existe une infinité d'hyperplans permettant de séparer parfaitement les données en deux groupes, comme l'illustre la figure 3.8.

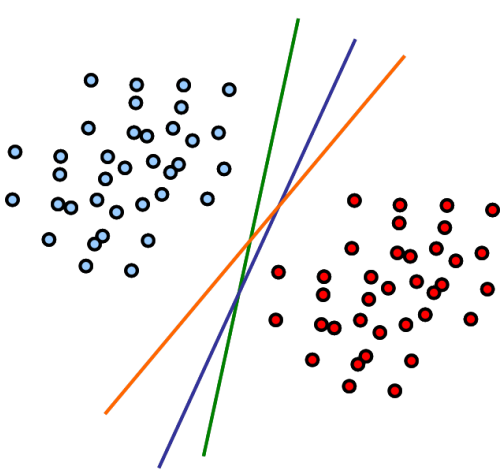


FIGURE 3.8 – 2 classes parfaitement séparables. Il existe une infinité d'hyperplans séparateurs.

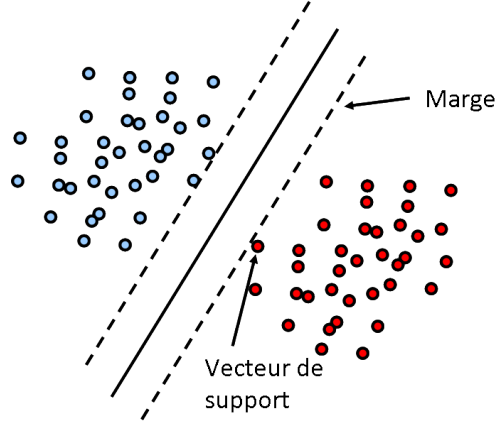


FIGURE 3.9 – 2 classes parfaitement séparables. L'hyperplan optimal est celui qui maximise la marge i.e. la distance entre l'hyperplan séparateur et les vecteurs de support.

Le SVM cherche alors à trouver l'hyperplan de séparation optimal au sens de la maximisation de la distance aux points les plus proches, appelés vecteurs de support, comme l'illustre la figure 3.9.

Dans de nombreux problèmes de classification (et en particulier en ICM), les classes ne sont pas systématiquement linéairement séparables, on peut néanmoins traiter ces cas avec le SVM grâce à l'introduction d'une fonction qui projette les données dans un espace de plus grande dimension où elles seront linéairement séparables. Le classifieur SVM est traité en détail en annexe C.

Les SVM sont couramment utilisés en ICM, car ils offrent de bonnes performances [44] et s'adaptent facilement à de nombreux problèmes et types d'ICM, que ce soit pour des potentiels évoqués [59], de l'imagerie motrice [110] ou encore pour la sélection d'électrodes [67]. Le SVM est également connu pour être robuste à l'augmentation de la dimension de l'espace de caractéristiques.

3.3.4 Régression logistique

cas binaire

La régression logistique est une technique de classification qui repose sur la modélisation des probabilités a posteriori des différentes classes. Dans un cas binaire ($Y \in \{0, 1\}$), cette modélisation prend la forme suivante :

$$Pr(Y = 1|\mathbf{v}_i) = \frac{\exp(\beta_0 + \boldsymbol{\beta}^T \mathbf{v}_i)}{1 + \exp(\beta_0 + \boldsymbol{\beta}^T \mathbf{v}_i)} \quad (3.37)$$

avec $Pr(Y = 1|\mathbf{v}_i) + Pr(Y = 0|\mathbf{v}_i) = 1$. Le problème de l'apprentissage revient à estimer le couple $(\beta_0, \boldsymbol{\beta})$ qui maximise la somme de la log-vraisemblance des échantillons. Cette problématique se résout par une procédure d'optimisation.

La classification s'effectue ensuite par :

$$\hat{y}_i = \begin{cases} 1 & \text{si } Pr(Y = 1|\mathbf{v}_i) > 0.5 \\ 0 & \text{sinon} \end{cases} \quad (3.38)$$

Extension multi-classe et régularisée

Il existe une généralisation à K classes, qui considère alors un ensemble de $K - 1$ problèmes binaires tel que :

$$\log \frac{Pr(Y = k|\mathbf{v}_i)}{Pr(Y = K|\mathbf{v}_i)} = \beta_{0,k} + \beta_k^T \mathbf{v}_i \quad (3.39)$$

avec $k \in \{1 : K - 1\}$ et sous la condition $\sum_{k=1}^K Pr(Y = k|\mathbf{v}_i) = 1$. On entraîne alors K couples $(\beta_{0,k}, \beta_k)$ et le calcul de la probabilité a posteriori, équation (3.40), devient :

$$Pr(Y = k|\mathbf{v}_i) = \frac{\exp(\beta_{0,k} + \beta_k^T \mathbf{v}_i)}{\sum_{l=1}^K \exp(\beta_{0,l} + \beta_l^T \mathbf{v}_i)} \quad (3.40)$$

La classification s'effectue par maximisation de la probabilité a posteriori :

$$\hat{y}_i = \underset{k}{\operatorname{argmax}} Pr(Y = k|\mathbf{v}_i) \quad (3.41)$$

Cette technique possède également une version régularisée [40], qui en ajoutant un terme de pénalisation dans la fonction d'optimisation permet de converger vers des solution parcimonieuses :

$$\{\beta_{0,k}, \beta_k\}_{k=[1:K]} = \underset{\{\beta_{0,k}, \beta_k\}_{k=[1:K]}}{\operatorname{argmax}} \left[\frac{1}{I} \sum_{i=1}^I \log Pr(Y = l|\mathbf{v}_i) - \lambda \sum_{l=1}^K P_\alpha(\beta_l) \right] \quad (3.42)$$

avec λ un paramètre de la régularisation et $P_\alpha(\cdot)$ une fonction telle que :

$$P_\alpha(\beta) = (1 - \alpha)\|\beta\|_2 + \alpha\|\beta\|_1 \quad (3.43)$$

Cette régularisation est appelée *elasticnet* et a été introduite dans [134]. Elle est constituée d'une pénalisation *Ridge regression* [53] couplée à la méthode *lasso* [115] qui effectue une sélection de variables en forçant certains coefficient à rester nuls.

3.3.5 Validation croisée

Pour l'évaluation des résultats et la sélection de paramètres, il est d'usage de recourir à des procédures de validations croisées (CV). Étant donné que l'entraînement des fonctions de classification (ou des filtres spatiaux) se fondent sur les données contenues dans la base d'apprentissage, on converge vers des solutions optimales pour les données d'apprentissage. Il est alors possible que les solutions obtenues ne soient pas généralisables à des nouvelles données n'appartenant pas à cette base, par exemple des données enregistrées après l'apprentissage des fonctions de classification.

Il est donc nécessaire d'évaluer les performances d'une chaîne de traitement sur des données qui ne font pas partie des données d'apprentissage. La procédure de validation croisée consiste à partitionner les données disponibles en deux sous-ensembles, une base d'apprentissage et une base de test. On entraîne ainsi les algorithmes sur la base d'apprentissage et on évaluera les performances sur la base de test. La figure 3.10 illustre ce principe.

De la même manière, pour éviter que les performances évaluées ne s'attachent trop aux données contenues dans la base de test, on utilise une procédure nommée k -fold CV. Cette

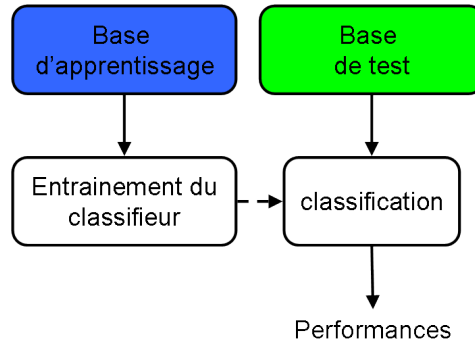


FIGURE 3.10 – Principe d'évaluation des performances, l'apprentissage et le test sont réalisés sur des données différentes de manière indépendante.

procédure divise les données disponibles en k partitions dont $k - 1$ seront utilisées comme base d'apprentissage tout en évaluant les performances sur la dernière. En permutant tour à tour les partitions utilisées comme base de test, on parcourt l'ensemble des données. Les performances globales sont alors constituées de la moyenne des performances évaluées sur chacune des bases de test. Cette procédure est illustrée figure 3.11.

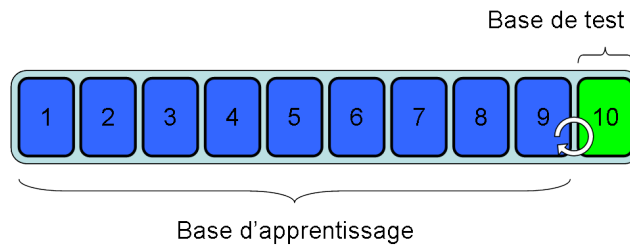


FIGURE 3.11 – 10-fold CV. On partitionne les données en 10 sous-ensembles, 9 sont utilisés comme base d'apprentissage et la dixième comme base de test. On permute 10 fois, la performance globale est la moyenne des performances individuelles sur chaque permutation.

Le rôle de cette permutation est d'obtenir en moyenne la même distribution sur les données d'apprentissage que sur celles de test.

3.4 Conclusions

Dans ce chapitre, les principales méthodes constituant la chaîne de traitement pour les ICMS basées sur l'imagerie motrice ont été présentées. La chaîne typique est principalement constituée de 4 blocs comme l'illustre la figure 3.12.

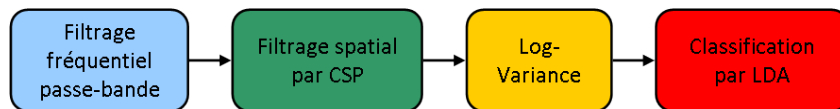


FIGURE 3.12 – Chaîne de traitement typique pour les ICMS basées sur de l'imagerie motrice.

- **Filtrage fréquentiel passe-bande** pour se limiter aux fréquences d'intérêt impliquées dans l'imagination des mouvements. Généralement entre 8 et 30 Hz.

- **Filtrage spatial par CSP** pour combiner les signaux dans l'optique d'en réhausser les caractéristiques spatiales. On utilise généralement 3 paires de filtres.
- **Extraction de caractéristiques** par log-variance pour représenter les réalisations de chaque tâche mentale à classer.
- **Classification par FLDA** pour déterminer le type de tâche mentale réalisée.

Cette chaîne de traitement se fonde majoritairement sur l'hypothèse que les signaux EEG suivent des **lois normales multivariées** conditionnées à la classe. Ainsi, la CSP se calcule par diagonalisation conjointe d'un jeu de matrices de covariance, ce qui est analogue à un problème de séparation de sources. De même l'utilisation de la log-variance comme caractéristique découle de la même hypothèse pour obtenir des distributions normales et appliquer une classification par LDA de manière efficace.

Finalement, toutes ces opérations permettent de représenter, à un haut niveau, les distributions que suivent les signaux EEG pendant la réalisation d'un mouvement imaginé.

Partant de cette hypothèse fondamentale, le chapitre suivant décrira des méthodes fondées sur la géométrie Riemannienne dans le but de manipuler directement les lois normales multivariées.

Deuxième partie

Contributions Algorithmiques

Géométrie Riemannienne :

Approche fondée sur la distance

4.1 Introduction

4.1.1 La chaîne de traitement CSP revisitée

Prise dans son ensemble, l'extraction de caractéristiques, présentée dans le chapitre 3, peut être vue comme un ensemble d'opérations visant à représenter les matrices de covariance estimées sur des portions de signal EEG dans un espace où elles seront classées par un algorithme de l'état de l'art. En effet, l'extraction de caractéristiques prend la forme suivante (3.19) :

$$\mathbf{v}_i = \log(\text{diag}(\mathbf{W}^T \boldsymbol{\Sigma}_i \mathbf{W})) \quad (4.1)$$

où \mathbf{W} est un ensemble de filtres spatiaux calculé par CSP et $\boldsymbol{\Sigma}_i$ est la matrice de covariance estimée sur la portion de signal EEG \mathbf{X}_i . La projection par CSP ramène l'ensemble des informations utiles à la discrimination des classes sur la diagonale de la matrice, qui peut donc être représentée sans trop de pertes par un vecteur. L'application du logarithme assure ensuite que les données suivent une loi proche d'une loi normale et facilite donc l'emploi d'un classifieur linéaire.

Le vecteur \mathbf{v}_i représente donc la matrice de covariance $\boldsymbol{\Sigma}_i$. Pour que cette représentation soit sans perte d'information sur la tâche mentale, il faut respecter deux conditions :

1. **Les filtres spatiaux sont efficaces.** Le rôle des filtres spatiaux est de séparer l'information pertinente (utile à la tâche de classification) du bruit. Si cette séparation n'est pas efficace, on perd une partie de l'information qui ne pourra donc pas être exploitée par le classifieur. En pratique, on sera confronté au problème de robustesse dans le calcul de ces filtres, qui est directement lié à la qualité des données de la base d'apprentissage. En particulier si les données sont corrompues par un artefact, les filtres calculés seront biaisés et les caractéristiques présentées au classifieur ne seront pas pertinentes pour la séparation des classes.
2. **La diagonalisation des matrices est parfaite.** Étant donné que l'on utilise uniquement la variance des signaux, l'information contenue dans les éléments hors diagonale sera perdue. La quantité d'information perdue dépend grandement de la répartition des données, et elle sera d'autant plus faible que les matrices $\boldsymbol{\Sigma}_i$ sont proches des matrices de covariance intra-classes $\boldsymbol{\Sigma}^{(k)}$.

Ces deux conditions n'étant jamais vérifiées dans la pratique, on s'attend donc à une sous-exploitation de l'information contenue dans les matrices de covariance. Partant de ces constatations, ce document présentera donc un ensemble de méthodes permettant d'exploiter au mieux les informations contenues dans les matrices de covariance. Le gain attendu est directement lié à la validité des deux conditions présentées dans cette section. On s'attend donc à un gain important dans le cas multi-classe (où la diagonalisation des matrices est moins efficace) et pour des données fortement bruitées (où les filtres ont tendance à être mal estimés).

4.1.2 Formulation du problème et approche suivie

On peut considérer les signaux EEG $\mathbf{x}(t)$ mesurés à chaque instant t comme étant la réalisation d'un processus aléatoire suivant une loi normale centrée¹ $\mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$ de matrice de covariance spatiale $\mathbf{\Sigma}$. En particulier, on peut considérer que le signal est stationnaire par morceaux et que la réalisation d'une tâche mentale induira une modification des paramètres de cette loi. En d'autres termes, quand l'utilisateur effectue une tâche mentale A, les signaux EEG suivent pendant tout le temps de la réalisation de la tâche mentale une loi $\mathcal{N}_A = \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_A)$ et quand il effectue une tâche mentale B, les signaux EEG suivent une autre loi $\mathcal{N}_B = \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_B)$. C'est cette approche qui est à la base de l'extension multiclasse de la CSP [47, 48] présentée section 3.2.4.

Suivant ce principe, la réalisation d'une tâche mentale est entièrement caractérisée par la matrice de covariance estimée sur la portion temporelle de signal EEG correspondante. La matrice de covariance peut donc **être utilisée comme descripteur du signal EEG**.

De plus on pourra noter que dans le cadre de l'imagerie motrice, outre le fait que le signal EEG est bruité et que ce bruit peut varier temporellement, l'utilisateur n'effectue pas de manière uniforme une même tâche mentale à travers ses diverses réalisations. Ainsi, dans notre cas, pour deux réalisations de la même tâche mentale A, les matrices de covariance estimées ne seront pas identiques et égales à $\mathbf{\Sigma}_A$ et sont donc la réalisation d'un processus aléatoire, en toute vraisemblance, suivant une distribution de Wishart paramétrée par $\mathbf{\Sigma}_A$. On pourra alors résumer notre problématique ainsi :

Hypothèse. Lors de la réalisation d'une tâche mentale de classe y_i démarrant au temps t_i et durant N_i échantillons temporels, les signaux EEG suivent alors une loi normale centrée $\mathcal{N}_i = \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_i)$, la matrice de covariance $\mathbf{\Sigma}_i$ étant la réalisation d'un processus aléatoire conditionné par la classe y_i .

Problématique. Pour découvrir le type de tâche mentale réalisée pendant une portion de signal EEG, notre approche consiste à estimer et classer la distribution \mathcal{N}_i estimée sur cette portion de signal. La distribution étant centrée, elle est uniquement paramétrée par la matrice de covariance. De manière équivalente, on cherche donc à estimer une fonction de classification h qui relie la matrice de covariance d'une réalisation à sa classe y_i i.e. au type de tâche mentale réalisée :

$$y_i = h(\mathbf{\Sigma}_i) \quad (4.2)$$

Afin de réaliser cette classification, il est donc nécessaire de définir des outils permettant de comparer ces matrices de covariance. D'une manière générale, classer des données revient à les comparer entre elles au sens d'une certaine métrique, à rechercher leurs similarités ou divergences. Il convient donc d'utiliser la bonne métrique en lien avec la nature des données. La section suivante introduira les concepts de la géométrie de l'information, domaine en pleine expansion, qui s'attache à appliquer une approche géométrique dans le cadre de la théorie de l'information.

4.1.3 Etat de l'art des approches similaires

Dans de nombreux domaines, les matrices de covariance sont utilisées dans des algorithmes très divers. Ces matrices de covariance sont symétriques et, sauf cas particuliers, définies positives (SPD). Elles appartiennent ainsi à une variété Riemannienne particulière. Toutefois, dans la plupart des cas, elles sont traitées avec des outils de la géométrie Euclidienne. La prise en compte de la topologie particulière de l'espace des matrices SPD permet de manipuler ces matrices de manière plus rigoureuse et apporte souvent une amélioration significative des performances. Les concepts de géométrie Riemannienne pour l'espace des

1. Grâce à un filtrage fréquentiel passe bande, les signaux EEG sont de moyenne nulle.

matrices de covariance n'ont, à notre connaissance, jamais été appliqués directement au domaine des ICMs. En revanche, ils sont de plus en plus utilisés pour tout ce qui a trait au traitement de signaux multivariés comme pour le traitement des signaux radar [13, 14], des images ou vidéo [102, 116] et même plus récemment sur des signaux EEG dans le contexte de la classification des phases de sommeil [72, 73].

Le suivi de trajectoire ou d'objet dans des séquences d'images ou des vidéos. Dans le cadre de ce qu'on appelle la *Vision par ordinateur*, on est amené à mettre au point des méthodes permettant le suivi de cibles dans les vidéos. Par exemple, on peut chercher à compter et à suivre le nombre de piétons dans une scène [102]. On définit alors un ensemble de caractéristiques (luminance, couleur, gradient, etc.) sur une portion d'image (une fenêtre de quelques pixels), on calcule une matrice de covariance à partir de ces caractéristiques. Cette matrice de covariance représente à un haut niveau d'abstraction le contenu de la portion d'image, et grâce à la métrique de l'espace des matrices SPD, on va pouvoir suivre l'évolution de la position de ces portions d'images au fil de la vidéo ou encore dénombrer le nombre d'objets similaires dans la scène [116].

Imagerie radar. Ces concepts sont utilisés en imagerie Doppler haute résolution [13, 14] pour la détection de petites cibles en imagerie radar. La géométrie Riemannienne, permet dans ce cas présent, de conserver la structure Toeplitz (spécifique à ces données d'imagerie Doppler) des matrices de covariance lors des calculs de moyenne de matrices de covariance nécessaires à la résolution de cette problématique de détection. En particulier, la détection de cibles est réalisée à partir d'un critère de distance Riemannienne à la matrice de covariance médiane.

Imagerie des tenseurs de diffusion en IRM : Les tenseurs de diffusion en IRM sont composés de matrices SPD de taille 3×3 pour chaque voxel. Si l'on souhaite extraire les formes principales de ces images, on est amené à utiliser une PCA. Le problème est que les lignes définies par les modes de variation (composantes principales) ne conservent pas le caractère SPD de chaque voxel [38]. Il est alors possible de définir une nouvelle version de la PCA dans l'espace des matrices SPD afin de respecter ces propriétés [39].

Classification de signaux EEG : Les concepts de la géométrie Riemannienne (GR) ont également été appliqués à la classification de signaux EEG dans le cadre de la détection des phases de sommeil [72, 73]. Une métrique est établie pour les matrices de densité spectrale de puissance (DSP). Une distance Riemannienne (pondérée) entre deux tenseurs de densité spectrale de puissance est ensuite définie puis utilisée pour classer des portions de signal EEG à l'aide d'un algorithme des plus proches voisins.

4.2 La géométrie de l'information

Nous allons, dans cette section, présenter les concepts essentiels de la géométrie de l'information et les appliquer à la variété des lois normales centrées et multivariées. Cette variété est également celle des matrices symétriques et définies positives, car ces lois sont uniquement paramétrées par la matrice de covariance. Les outils de la géométrie Riemannienne dans la variété des matrices de covariance peuvent être définis de manière identique à partir des propriétés algébriques de l'espace des matrices SPD. Nous présenterons ici la vision empruntée par la géométrie de l'information, dont l'application à notre contexte (modélisation des signaux EEG par des lois centrées normales) semble plus naturelle.

4.2.1 Variété des distributions de probabilité

La géométrie de l'information considère que chaque famille de distribution de probabilité peut être vue comme une variété, notée \mathcal{M} . L'un des systèmes de coordonnées le plus naturel pour cette variété peut être constitué des paramètres de la distribution en question. Par exemple, pour les distributions gaussiennes unidimensionnelles $\mathcal{N}(\mu, \sigma^2)$, le système de coordonnées est formé par la moyenne μ et la variance σ^2 .

En d'autres mots, chaque distribution gaussienne de paramètre μ et σ^2 est un point de coordonnées (μ, σ^2) de la variété des distributions de probabilité gaussienne. Ce principe est schématisé par la figure 4.1

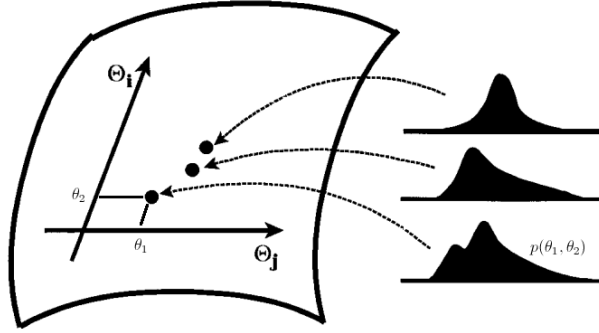


FIGURE 4.1 – Chaque distribution de probabilité d'une même famille est un point de la variété. Les coordonnées de ce point sont les valeurs que prennent les paramètres de la distribution dans le système de coordonnées formé des paramètres libres de la famille de distribution en question. Ici on représente 3 distributions d'une famille notée $p(\Theta_i, \Theta_j)$, dont l'une, notée $p(\theta_1, \theta_2)$, a pour coordonnées le couple (θ_1, θ_2)

Une variété est un espace topologique (courbe) qui peut être localement approximé par un espace euclidien. Pour prendre un exemple, une mappemonde est une variété que l'on approxime localement à l'aide de cartes 2D se chevauchant.

De plus, cette variété des distributions de probabilité est différentiable c'est-à-dire que l'on peut y mener des calculs différentiels. Si elle est munie d'une métrique, c'est une variété Riemannienne différentiable.

4.2.2 Plan tangent, géodésique et distance

Une variété Riemannienne possède en tout point \mathbf{P} un plan tangent $\mathcal{T}_{\mathbf{P}}\mathcal{M}$ muni d'un produit scalaire $\langle \cdot, \cdot \rangle_{\mathbf{P}}$ qui permet la définition de la métrique. Ce plan tangent est Euclidien et également localement homéomorphe à la variété. Le plan tangent peut être vu comme étant constitué de l'ensemble des vecteurs tangents au point \mathbf{P} partant dans toutes les directions, comme l'illustre la figure 4.2.

Le produit scalaire dans le plan tangent induit la définition d'une norme. Soit \mathbf{S} un élément du plan tangent, on a :

$$\|\mathbf{S}\|_{\mathbf{P}} = \sqrt{\langle \mathbf{S}, \mathbf{S} \rangle_{\mathbf{P}}} \quad (4.3)$$

On considère maintenant une courbe paramétrée par le réel $t \in [0, 1]$ sur la variété, et notée α . Pour chaque $\alpha(t)$ de la variété, on a donc un vecteur tangent $\dot{\alpha}(t)$ dont la norme est $\|\dot{\alpha}(t)\|_{\alpha(t)}$. La longueur de la courbe partant de $\alpha(0)$ et arrivant en $\alpha(1)$ est alors définie à partir de l'intégrale suivante :

$$L(\alpha) = \int_0^1 \|\dot{\alpha}(t)\|_{\alpha(t)} dt \quad (4.4)$$

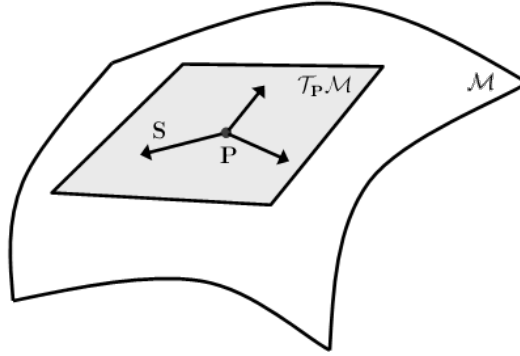


FIGURE 4.2 – Le plan tangent $\mathcal{T}_P \mathcal{M}$ est défini localement, au point P , sur la variété \mathcal{M} . Le plan tangent $\mathcal{T}_P \mathcal{M}$ est constitué de l'ensemble des vecteurs tangents S partant de P .

On nomme alors **géodésique** la courbe dont la longueur est minimale :

$$\Gamma = \operatorname{argmin}_{\alpha} L(\alpha) \quad (4.5)$$

Cette longueur définit alors la **distance** entre deux points de la variété. Soient 2 points de la variété P_1 et P_2 , soit Γ la géodésique paramétrée par t et partant de $\Gamma(0) = P_1$ vers $\Gamma(1) = P_2$ et $\Gamma(t)$ un point de cette géodésique, on définit la distance Riemannienne $\delta_R(.,.)$ par :

$$\delta_R(P_1, P_2) = L(\Gamma) = \int_0^1 \|\dot{\Gamma}(t)\|_{\Gamma(t)} dt, \quad (4.6)$$

La figure 4.3 schématise ce principe.

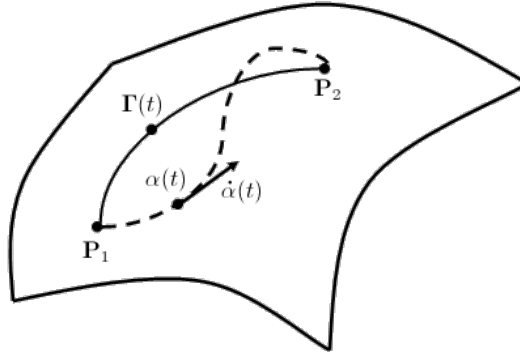


FIGURE 4.3 – La courbe quelconque α (en pointillés) reliant deux points P_1 et P_2 de la variété. $\alpha(t)$ un point de cette courbe et $\dot{\alpha}(t)$ son vecteur tangent. La géodésique Γ (courbe de longueur minimale) et un point $\Gamma(t)$ de cette géodésique. La distance entre P_1 et P_2 se fait par intégration de la norme du vecteur tangent $\dot{\Gamma}(t)$ le long de la géodésique.

4.2.3 Métrique

La métrique d'une variété Riemannienne est une fonction qui assigne à chaque point P de cette variété un produit scalaire $\langle ., . \rangle_P$ dans l'espace tangent. La métrique est donc à la base de la formulation du produit scalaire et donc de la distance. En géométrie de l'information, **l'information de Fisher est considérée comme la métrique naturelle** des variétés

de distribution de probabilité. L'information de Fisher quantifie l'information relative à un paramètre contenue dans une distribution. La métrique étant la fonction permettant de quantifier la distance qui sépare deux éléments de la variété et la variété \mathcal{M} ayant comme système de coordonnées les paramètres de la distribution, le choix de l'information de Fisher comme métrique apparaît comme évident et fournit donc une distance qui quantifie la différence d'information entre des distributions de paramètres différents.

Soit une variété de dimension m , on appelle également tenseur métrique le tenseur d'ordre 2, représenté sous la forme d'une matrice $m \times m$ SPD, dont les éléments sont le produit scalaire des vecteurs de base du plan tangent. Ainsi :

$$\mathbf{G} = [G_{i,j}] \text{ avec } G_{i,j} = \langle \mathbf{e}_i, \mathbf{e}_j \rangle_{\mathbf{P}} \quad (4.7)$$

avec \mathbf{e}_i et \mathbf{e}_j deux vecteurs de base du plan tangent. Ce tenseur métrique est donc défini pour chaque point \mathbf{P} de la variété.

On a alors le déplacement infinitésimal sur une ligne de la variété qui est défini par :

$$ds^2 = \sum_{i,j}^m G_{i,j} d\theta_i d\theta_j = d\boldsymbol{\theta}^T \mathbf{G} d\boldsymbol{\theta} \quad (4.8)$$

C'est en quelque sorte la distance entre un point \mathbf{P} de la variété de coordonnées $\{\theta_1, \dots, \theta_i, \dots, \theta_m\}$ et son voisin immédiat \mathbf{P}' de coordonnées $\{\theta_1 + d\theta_1, \dots, \theta_j + d\theta_j, \dots, \theta_m + d\theta_m\}$. On nomme cette quantité *line element* ou encore forme quadratique différentielle, mais on la retrouve également sous le nom de métrique dans une partie de la littérature.

La longueur d'une courbe sur la variété, donnée par l'équation (4.4), peut également être obtenue par intégration de cette quantité sous la contrainte de déplacement le long de la courbe.

Dans le cadre de la variété des distributions de probabilité, on obtient alors comme tenseur métrique la matrice d'information de Fisher. Soit $p(\mathbf{x}, \boldsymbol{\theta})$ une loi de probabilité, $\boldsymbol{\theta} = [\theta_1, \dots, \theta_i, \dots, \theta_m]$ le vecteur de m paramètres de cette loi et \mathbf{x} une réalisation, la métrique issue de l'information de Fischer est donnée par :

$$G_{i,j} = \mathbb{E} \left[\frac{\partial \log p(\mathbf{x}, \boldsymbol{\theta})}{\partial \theta_i} \frac{\partial \log p(\mathbf{x}, \boldsymbol{\theta})}{\partial \theta_j} \right] \quad (4.9)$$

4.3 Application aux distributions normales multivariées

Cette section appliquera ces principes aux distributions normales centrées multivariées $\mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$, ce qui nous permettra de mener à bien notre tâche de classification.

4.3.1 Métrique

Dans le cas d'une distribution normale multivariée $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ avec $\boldsymbol{\mu} \in \mathbb{R}^n$ le vecteur des moyennes et $\boldsymbol{\Sigma} \in \mathbb{R}^{n \times n}$ la matrice de covariance, les éléments de la matrice de Fisher, et donc le tenseur métrique prend la forme suivante [112] :

$$G_{i,j} = \frac{\partial \boldsymbol{\mu}^T}{\partial \theta_i} \boldsymbol{\Sigma}^{-1} \frac{\partial \boldsymbol{\mu}}{\partial \theta_j} + \frac{1}{2} \text{Tr} \left(\boldsymbol{\Sigma}^{-1} \frac{\partial \boldsymbol{\Sigma}}{\partial \theta_i} \boldsymbol{\Sigma}^{-1} \frac{\partial \boldsymbol{\Sigma}}{\partial \theta_j} \right) \quad (4.10)$$

En particulier, quand la distribution est centrée ($\boldsymbol{\mu} = \mathbf{0}$), on obtient :

$$G_{i,j} = \frac{1}{2} \text{Tr} \left(\boldsymbol{\Sigma}^{-1} \frac{\partial \boldsymbol{\Sigma}}{\partial \theta_i} \boldsymbol{\Sigma}^{-1} \frac{\partial \boldsymbol{\Sigma}}{\partial \theta_j} \right) \quad (4.11)$$

La variété Riemannienne correspondante est de dimension $m = \frac{n(n+1)}{2}$ car elle est paramétrée uniquement par la matrice de covariance Σ qui est une matrice symétrique. La matrice de covariance étant également définie positive, cette variété est également celle des matrices symétriques et définies positives (SPD), que l'on note généralement $P(n)$.

La forme quadratique étant définie par $ds^2 = \sum_{i,j} G_{ij} d\theta_i d\theta_j$ on a alors, par linéarité de la trace :

$$ds^2 = \frac{1}{2} \text{Tr} \left(\Sigma^{-1} \left(\sum_i \frac{\partial \Sigma}{\partial \theta_i} d\theta_i \right) \Sigma^{-1} \left(\sum_j \frac{\partial \Sigma}{\partial \theta_j} d\theta_j \right) \right) \quad (4.12)$$

en notant $d\Sigma = \sum_i \frac{\partial \Sigma}{\partial \theta_i} d\theta_i$ on obtient

$$ds^2 = \frac{1}{2} \text{Tr} (\Sigma^{-1} d\Sigma \Sigma^{-1} d\Sigma) = \frac{1}{2} \|\Sigma^{-1} d\Sigma\|_F^2 = \frac{1}{2} \|\Sigma^{-1/2} d\Sigma \Sigma^{-1/2}\|_F^2 \quad (4.13)$$

avec $\|\cdot\|_F$ la norme de Frobenius.

Soient \mathbf{S}_1 et \mathbf{S}_2 deux matrices symétriques de taille $n \times n$, qu'on nomme également vecteurs² tangents, on obtient à partir de l'équation (4.11) la formulation du produit scalaire dans l'espace tangent. On pourra raisonnablement supprimer au passage le facteur $\frac{1}{2}$.

$$\langle \mathbf{S}_1, \mathbf{S}_2 \rangle_\Sigma = \text{Tr} (\mathbf{S}_1 \Sigma^{-1} \mathbf{S}_2 \Sigma^{-1}) \quad (4.14)$$

Le produit scalaire permet alors la définition d'une norme pour les vecteurs tangents : $\|\mathbf{S}\|_\Sigma^2 = \langle \mathbf{S}, \mathbf{S} \rangle_\Sigma = \text{Tr} (\Sigma^{-1} \mathbf{S} \Sigma^{-1})$. On pourra remarquer que cette norme se simplifie pour donner la norme de Frobenius lorsque le plan tangent est calculé au niveau de la matrice identité, c'est-à-dire $\langle \mathbf{S}, \mathbf{S} \rangle_{\mathbf{I}} = \|\mathbf{S}\|_F^2$.

Une autre propriété de cette métrique est qu'elle est invariante par projection congruente c'est-à-dire pour toute matrice $\mathbf{W} \in Gl(n)$ inversible, on a :

$$\langle \mathbf{W} \mathbf{S}_1 \mathbf{W}^T, \mathbf{W} \mathbf{S}_2 \mathbf{W}^T \rangle_{\mathbf{W} \Sigma \mathbf{W}^T} = \langle \mathbf{S}_1, \mathbf{S}_2 \rangle_\Sigma \quad (4.15)$$

4.3.2 Géodésique et distance

Soit $\Gamma(t) : [0, 1] \rightarrow P(n)$ la géodésique allant de $\Gamma(0) = \Sigma_1$ vers $\Gamma(1) = \Sigma_2$. La longueur de $\Gamma(t)$ alors donnée par :

$$L(\Gamma) = \int_0^1 \|\dot{\Gamma}(t)\|_{\Gamma(t)} dt = \int_0^1 \sqrt{\text{Tr} (\dot{\Gamma}(t) \Gamma^{-1}(t) \dot{\Gamma}(t) \Gamma^{-1}(t))} dt, \quad (4.16)$$

On remarquera que cette distance hérite également de la part de la métrique de la propriété d'invariance par projection, c'est à dire $L(\Gamma) = L(\mathbf{W} \Gamma \mathbf{W}^T)$

La distance entre Σ_1 et Σ_2 est obtenue par le calcul de cette intégrale. Il est alors nécessaire de calculer la formule de la géodésique au sens de la métrique en question. L'annexe D.1 présente la démonstration permettant d'obtenir la formule de cette géodésique. On obtient ainsi :

$$\Gamma(t) = \Sigma_1^{\frac{1}{2}} \left(\Sigma_1^{-\frac{1}{2}} \Sigma_2 \Sigma_1^{-\frac{1}{2}} \right)^t \Sigma_1^{\frac{1}{2}}. \quad (4.17)$$

L'application de cette géodésique à l'équation (4.16) permet finalement d'obtenir la formule de la distance :

$$\delta_R(\Sigma_1, \Sigma_2) = \|\log (\Sigma_1^{-1} \Sigma_2)\|_F = \left[\sum_{i=1}^n \log^2 \lambda_i \right]^{1/2}, \quad (4.18)$$

2. Par abus de langage, ces matrices sont appelées vecteurs tangents, ce qui correspond au cas général où les éléments d'une variété ne sont pas nécessairement représentés par une matrice.

avec $\{\lambda_i\}_{i \in \{1:n\}}$ les valeurs propres réelles de $\Sigma_1^{-1}\Sigma_2$. Les principales propriétés de cette distance sont :

- $\delta_R(\Sigma_2, \Sigma_1) = \delta_R(\Sigma_1, \Sigma_2)$
- $\delta_R(\Sigma_1^{-1}, \Sigma_2^{-1}) = \delta_R(\Sigma_1, \Sigma_2)$
- $\delta_R(\mathbf{W}^T \Sigma_1 \mathbf{W}, \mathbf{W}^T \Sigma_2 \mathbf{W}) = \delta_R(\Sigma_1, \Sigma_2) \forall \mathbf{W} \in Gl(n)$.

Cette distance est également appelée *Affine-invariante* [4], car elle est définie à partir de la géodésique affine et la propriété d'invariance par projection de la métrique.

Il existe d'autres approches permettant de définir une distance entre deux matrices de covariance. L'annexe E présente 3 autres distances couramment utilisées dans la littérature. Il s'agit de la distance Euclidienne, de la distance de Kullback-Leibler définie à partir de la divergence de Kullback-Leibler pour les distributions centrées normales et de la distance Log-Euclidienne. La distance Riemannienne présentée ici sera comparée à ces trois autres distances dans la suite de ce chapitre.

4.3.3 Moyenne Riemannienne

De nombreux algorithmes de classification utilisent la moyenne d'un ensemble de données. Cette notion de moyenne n'est pas triviale lorsqu'on se place dans une variété. En utilisant la distance Riemannienne (4.18), et par analogie avec la moyenne Euclidienne, on définit la moyenne Riemannienne de $I \geq 1$ matrices SPD par :

$$\mathfrak{G}(\Sigma_1, \dots, \Sigma_I) = \underset{\Sigma \in P(n)}{\operatorname{argmin}} \sum_{i=1}^I \delta_R^2(\Sigma, \Sigma_i) \quad (4.19)$$

Cette moyenne est également appelée barycentre de Frechet ou encore **moyenne géométrique**. Pour une variété à courbure négative comme $P(n)$, un tel minimum local existe et il est unique [62]. Cependant il n'y a pas de formule littérale pour calculer cette moyenne et une procédure d'optimisation doit être employée. Les détails de cette procédure sont donnés en annexe D.2.

4.4 Classification dans la variété Riemannienne

4.4.1 Introduction

La géométrie de l'information a permis de définir la notion de distance entre deux matrices de covariance, ainsi que la notion de moyenne liée à cette définition de la distance. Nous allons maintenant pouvoir utiliser ces concepts pour classer directement les matrices de covariance spatiales en EEG.

Il existe un grand nombre d'algorithmes de classification se fondant sur une comparaison de distances. Les plus connus sont les algorithmes des k plus proches voisins (*k-Nearest Neighbor* : k-NN), du minimum de distance à la moyenne (*Minimum Distance to Mean* : MDM) ou encore l'algorithme des k moyennes (*k-means*).

4.4.2 k plus proches voisins

L'algorithme des k plus proches voisins se fonde uniquement sur des calculs de distance. L'algorithme peut être résumé ainsi : la classe d'un élément est égale à la classe qui est majoritaire parmi ces k voisins. La Figure 4.4 illustre ce processus. L'algorithme k -NN est supervisé. On dispose donc de I matrices de covariance Σ_i de classe respective y_i . Lorsqu'on souhaite classer une réalisation de classe inconnue y et de matrice de covariance Σ , l'algorithme s'effectue en 2 étapes :

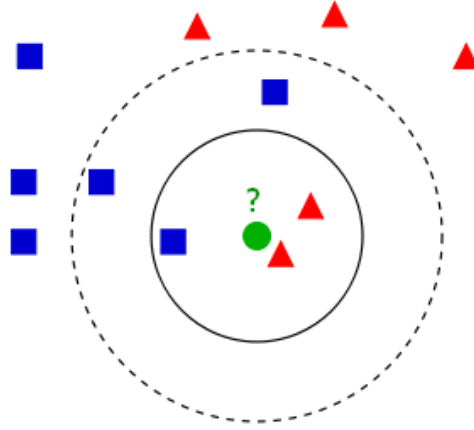


FIGURE 4.4 – Illustration du principe de l'algorithme des k plus proches voisins. Avec $k = 3$ le rond vert sera classé en tant que triangle rouge alors que pour $k = 5$ il sera classé en tant que carré bleu.

1. Recherche des voisins : on calcule les I distances $\delta(\Sigma, \Sigma_i)$, on les trie par ordre croissant. Les k voisins sont alors les k points dont la distance est minimale.
2. Classification : pour les k voisins sélectionnés, on compte le nombre d'éléments de chaque classe. On affecte alors à y la classe la plus fréquente.

L'algorithme équivalent est donné en Algorithme 1.

Algorithme 1 Classification par algorithme k-NN

Entrée : Ω un ensemble de I matrices de covariance $\Sigma_i \in P(n)$.

Entrée : K le nombre de classes.

Entrée : $y_i \in [1 : K]$ la classe de Σ_i .

Entrée : k le nombre de voisins.

Entrée : Σ une matrice de covariance de classe inconnue.

Sortie : \hat{y} l'estimation de la classe de la matrice de covariance à tester Σ .

```

1: for  $i = 1$  to  $I$  do
2:    $D(i) = \delta(\Sigma, \Sigma_i)$  {Calcul des distances}
3: end for
4:  $[\prec, ix] = \text{sort}(D)$  {tri des tests par ordre croissant de distance}
5:  $Voisins = ix(1 : k)$  {sélection des k plus proches voisins}
6: for  $c = 1$  to  $K$  do
7:    $Nb(c) = \text{Compte}(y_i = c | i \in Voisins)$  {Compte le nombre de voisins de chaque classe}
8: end for
9:  $\hat{y} = \text{argmax}_c(Nb(c))$  {Maximum de voisins de même classe}
10: return  $\hat{y}$ 

```

Il est possible que pour un nombre de k voisins, on retrouve un nombre égal de voisins pour deux classes différentes. Il y a alors une indétermination sur la classe. Dans le cas où le nombre de classes est $K = 2$, il suffit simplement d'utiliser un nombre impair de voisins k pour ne jamais entrer dans ce cas de figure. Pour $K > 2$, on pourra également lever cette indétermination en sélectionnant la classe dont la somme des distances est la plus faible.

Enfin, il n'est pas rare que cet algorithme soit utilisé en combinaison avec une distance pondérée [73] qui minimisera de manière artificielle la distance entre les points d'une même

classe.

L'avantage de cet algorithme réside dans sa **simplicité**, mais aussi dans son **efficacité** en particulier dans le cas où le nombre I de points dans la base d'apprentissage est grand. En revanche, il s'avère relativement coûteux en ressources, d'une part car il faut conserver en mémoire l'ensemble de la base d'apprentissage pour classer un point, et d'autre part car il faut calculer une distance pour chaque point de la base d'apprentissage ce qui peut s'avérer long quand I est grand.

La figure 4.5 illustre un cas de classification utilisant cet algorithme pour les 4 distances (la distance Riemannienne présentée section 4.3.2 et les 3 distances présentées annexe E). Cette courbe montre l'évolution du pourcentage du nombre de voisins correspondant à la classe du point que l'on cherche à classer. Ainsi, un pourcentage de 80% indique que parmi l'ensemble des k voisins, 0.8 k d'entre eux sont de la classe du point considéré. Plus ce pourcentage sera élevé, plus la classification sera efficace.

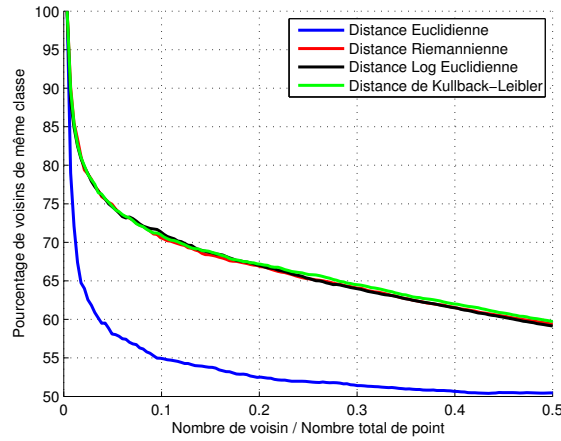


FIGURE 4.5 – Évolution du pourcentage de voisins de même classe en fonction de k/I (rapport entre k et le nombre total de points dans la base d'apprentissage I). Jeu de données Ila, BCI compétition III, Utilisateur 1.

On remarque ainsi que les distances Riemannienne, Log Euclidienne et de Kullback-Leibler sont globalement équivalentes pour cet algorithme. En revanche, la distance Euclidienne s'avère beaucoup moins efficace.

4.4.3 Distance minimale aux centres

L'algorithme de distance minimale aux centres (MDM) utilise l'estimation du centre des classes pour effectuer la classification. On peut résumer l'algorithme ainsi : la classe d'un élément est égale à la classe du centre plus proche. Ici l'estimation du centre pourra s'effectuer par la moyenne ou la médiane de l'ensemble des matrices d'une même classe.

Ainsi, l'algorithme comporte deux étapes :

1. Estimation des centres des classes : si on dispose de K classes, alors on estime le centre de chaque classe par un calcul de moyenne ou de médiane. On note $\Sigma^{(k)}$ le centre de la classe d'indice $k = [1 : K]$.
2. Classification : on affecte au point Σ de classe inconnue la classe du centre le plus proche : $y = \operatorname{argmin}_k \delta(\Sigma, \Sigma^{(k)})$

Cet algorithme est présenté en algorithme 2.

Algorithm 2 Classification par distance aux centres

 Entrée : Ω un ensemble de I matrices de covariance $\Sigma_i \in P(n)$.

 Entrée : $y_i \in [1 : K]$ la classe de Σ_i .

 Entrée : K le nombre de classes.

 Entrée : Σ une matrice de covariance de classe inconnue.

 Sortie : \hat{y} l'estimation de la classe de la matrice de covariance à tester Σ .

```

1: for  $k = 1$  to  $K$  do
2:    $\Sigma^{(k)} = \mathfrak{G}(\Sigma_i | y_i = k)$  {Calcul de la matrice de covariance moyenne pour la classe k}
3: end for
4:  $\hat{y} = \operatorname{argmin}_k (\delta(\Sigma, \Sigma^{(k)}))$  {Recherche de la distance minimale}
5: return  $\hat{y}$ 

```

En comparaison de l'algorithme des plus proches voisins, ce dernier est beaucoup plus léger en termes de calcul et d'occupation mémoire. En effet il ne nécessite la mémorisation que de K éléments (les centres des classes) et le calcul de K distances pour la classification. En revanche, les frontières de séparation entre chaque couple de classe sont des droites (au sens de la métrique) et forment donc des cellules de Voronoï comme l'illustre la figure 4.6. L'efficacité de la classification peut donc varier grandement en fonction de la distribution des données.

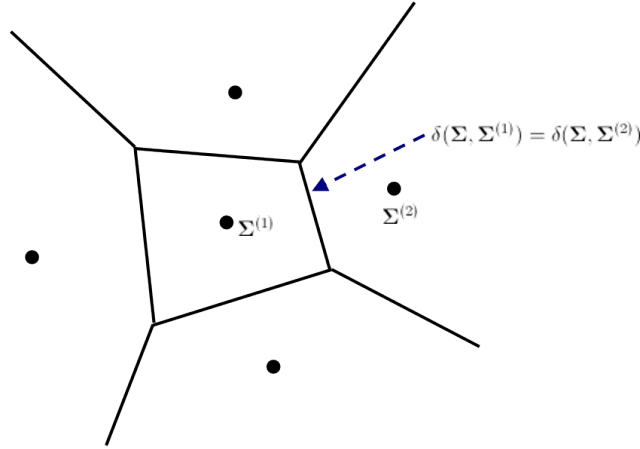


FIGURE 4.6 – Cellules de Voronoï. Les frontières de séparation entre les couples de classe sont des droites équidistantes des centres de chaque classe.

La figure 4.7 illustre ce processus de classification pour les quatre distances définies dans la section 4.3, sur des données réelles dans une ICM à deux classes (mouvement imaginé de la main droite par rapport au mouvement imaginé de la main gauche). On remarquera que la distance Riemannienne et la distance Log Euclidienne offrent une répartition des points relativement similaire. La distance Euclidienne, quant à elle, se montre inapte à séparer les deux classes. Pour la distance de Kullback-Leibler, l'algorithme est appliqué en combinaison avec la moyenne géométrique Riemannienne, car à notre connaissance la moyenne suivant la pseudo-distance de Kullback-Leibler n'est pas encore définie. On notera une distribution relativement différente de celle de la distance Riemannienne, avec néanmoins un pouvoir de séparation qui semble similaire.

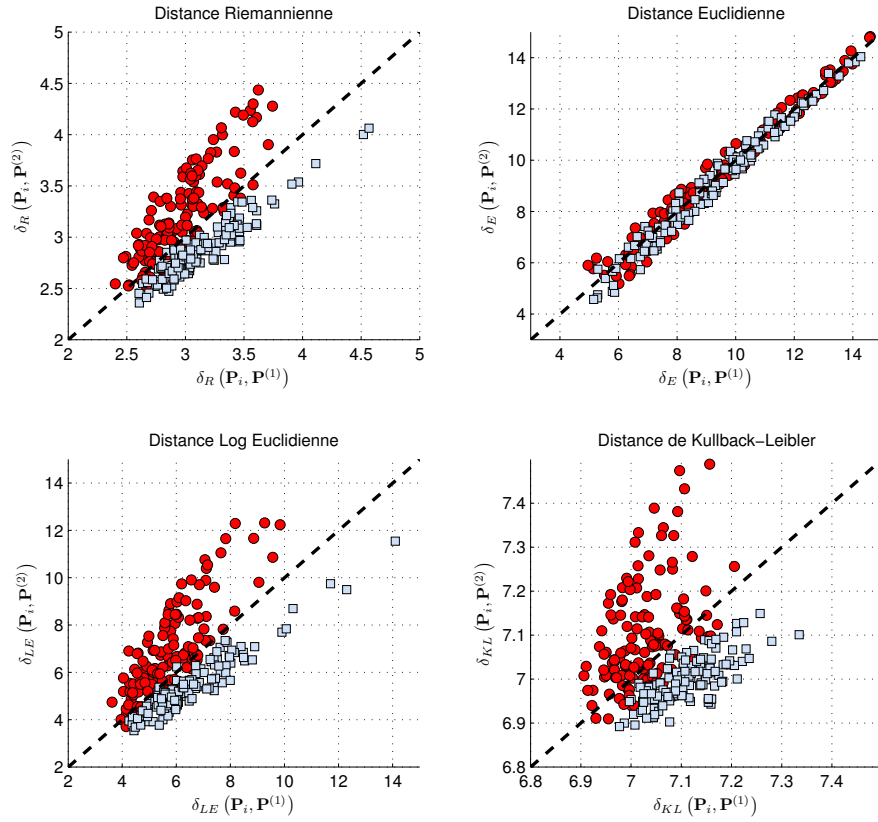


FIGURE 4.7 – Classification par minimum de distance aux centres, pour les différentes distances. Les cercles rouges correspondant à la classe (1) et les carrés bleus à la classe (2). En abscisse, la distance $\delta(\Sigma_i, \Sigma^{(1)})$ de chaque matrice à la matrice moyenne $\Sigma^{(1)}$ de la classe (1). En ordonnée la distance $\delta(\Sigma_i, \Sigma^{(2)})$ à la moyenne de la classe (2). La ligne en pointillé représente la fonction séparatrice d'équation $\delta(\Sigma_i, \Sigma^{(1)}) = \delta(\Sigma_i, \Sigma^{(2)})$

4.4.4 K-means

Les deux algorithmes présentés précédemment sont supervisés (cf section 3.3), c'est-à-dire qu'il est nécessaire d'avoir la connaissance des classes de chaque élément pour mener à bien l'apprentissage. L'algorithme des K -means est un algorithme non-supervisé. D'une manière plus générale, l'algorithme des K -means est un algorithme de partitionnement car il permet de regrouper un ensemble d'éléments en K classes. La classification ultérieure découle alors de ce partitionnement.

Sur le principe, le K -means est similaire à l'algorithme de la distance minimale aux centres. En effet la classification est également effectuée par minimisation de la distance aux centres de chaque classe. En revanche, ne connaissant pas les étiquettes de chaque matrice, l'estimation des centres est plus compliquée et passe par une procédure itérative.

Soient $\Omega^{(k)}$ l'ensemble des points attribués à la classe k et $\Sigma^{(k)}$ le centre de cet ensemble. Le partitionnement est dirigé par la minimisation itérative de la somme des carrés des

distances aux centres :

$$\min \sum_{k=1}^K \sum_{\Sigma_i \in \Omega^{(k)}} \delta(\Sigma_i, \Sigma^{(k)})^2 \quad (4.20)$$

L'algorithme se décompose suivant les étapes suivantes :

1. **Initialisation** : on initialise l'algorithme avec des valeurs arbitraires pour les K centres de classes $\Sigma_{t=0}^{(k)}$.
2. **Partitionnement** : à partir des centres, on en déduit un nouveau partitionnement : $y_{i,t} = \operatorname{argmin}_k \delta(\Sigma_i, \Sigma_t^{(k)})$
3. **Mise-à-jour des centres** : à partir du partitionnement, on estime le centre de chaque classe par le calcul de la moyenne géométrique : $\Sigma_{t+1}^{(k)} = \mathfrak{G}(\Sigma_i | y_{i,t} = k)$
4. **Arrêt** : si les centres des classes ont changé, on recommence à l'étape 2, sinon la procédure s'arrête.

Une fois cette procédure terminée, on obtient à la fois le partitionnement de la base d'apprentissage et les centres de chaque classe. Il suffit ensuite de classer tout point Σ de classe inconnue par MDM : $y = \operatorname{argmin}_k \delta(\Sigma, \Sigma_t^{(k)})$. Cet algorithme est donné en Algorithme 3.

Algorithm 3 Classification non-supervisée par K-means

Entrée : Ω un ensemble de I matrices de covariance $\Sigma_i \in P(n)$.

Entrée : K le nombre de classes.

Entrée : Σ une matrice de covariance de classe inconnue.

Sortie : Les centres des K classes $\Sigma^{(k)}$.

Sortie : \hat{y} l'estimation de la classe de la matrice de covariance à tester Σ .

```

1: initialisation : On choisit des valeurs pour les  $\Sigma_{t=0}^{(k)}$ 
2: repeat
3:   for  $i = 1$  to  $I$  do
4:      $y_{i,t} = \operatorname{argmin}_k \delta(\Sigma_i, \Sigma_t^{(k)})$  {partitionnement}
5:   end for
6:   for  $k = 1$  to  $K$  do
7:      $\Sigma_{t+1}^{(k)} = \mathfrak{G}(\Sigma_i | y_{i,t} = k)$  {Moyenne géométrique de la classe  $k$ }
8:   end for
9: until  $\Sigma_{t+1}^{(k)} = \Sigma_t^{(k)}$  pour tout  $k \in [1 : K]$ 
10:  $\hat{y} = \operatorname{argmin}_k \delta(\Sigma, \Sigma_t^{(k)})$  {Classification}
11: return  $\{\hat{y}, \Sigma_t^{(k)}\}$ 

```

Cet algorithme peut être très efficace si les différentes classes sont relativement bien séparées dans l'espace des caractéristiques. Ce n'est généralement pas le cas en ICM. Pour obtenir un espace de caractéristiques plus apte à la classification non supervisée, il faut généralement passer par une étape de filtrage spatial comme la CSP, qui est une étape supervisée. C'est la raison pour laquelle les techniques de classification non supervisée sont très peu utilisées en ICM. L'utilisation de la géométrie Riemannienne qui permet d'obtenir un espace de caractéristiques de qualité sans passer par une étape de filtrage spatial permet ainsi d'envisager l'utilisation de la classification non supervisée en ICM.

La figure 4.8 montre le partitionnement ainsi obtenu pour un cas où les données sont facilement séparables (tâche main gauche vs. pied).

La comparaison avec le partitionnement original (figure de gauche - MDM) montre que le partitionnement obtenu par la méthode non supervisée (figure de gauche - K-means) est équivalent. La performance en classification est de 96,5%.

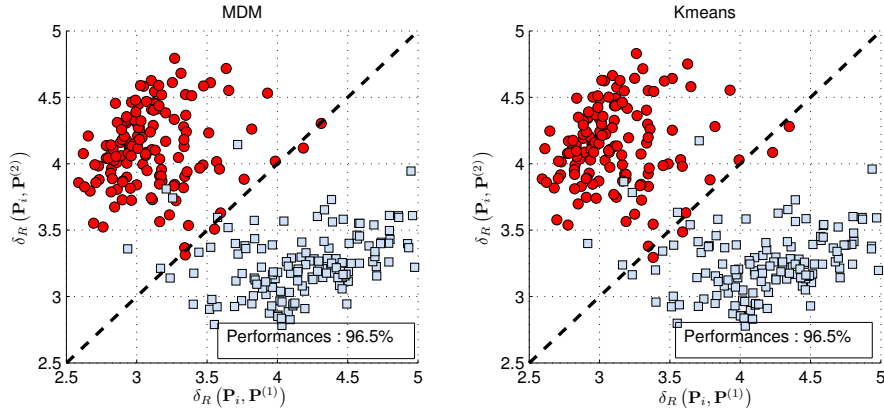


FIGURE 4.8 – k -means dans un cas très favorable. Le partitionnement original (figure de gauche - MDM) et le partitionnement obtenu par k -means (à droite) sont équivalents.

En revanche, nous avons observé que dans certains cas, les données sont séparables en deux groupes, mais que ces deux groupes ne correspondent pas aux deux classes. C'est le cas présenté figure 4.9, pour le même utilisateur mais pour deux tâches d'imagerie motrice différentes (langue vs. pied). Bien que les performances avec le partitionnement initial (MDM) soient bonnes (85.5%), le partitionnement obtenu par k -means sépare les données en deux groupes distincts ne correspondant pas aux deux classes. Dans le cas présent il s'avère que le partitionnement obtenu sépare les données suivant leur appartenance à une des deux sessions d'enregistrement dont le jeu de données est composé.

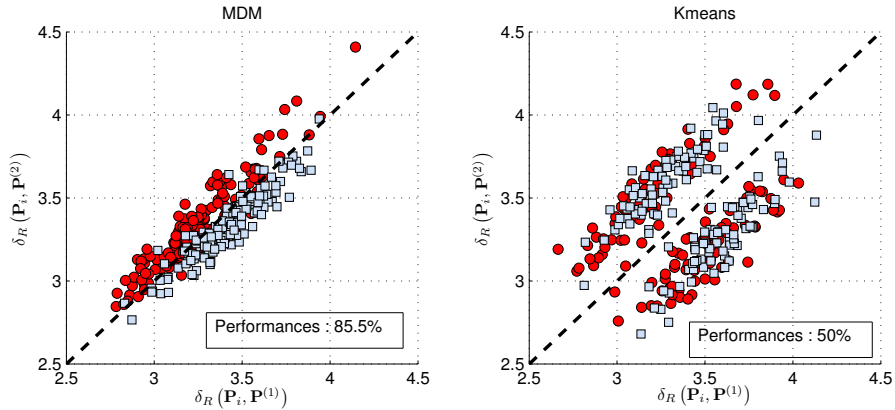


FIGURE 4.9 – k -means dans un cas peu favorable. Le partitionnement par k -means sépare parfaitement les données en deux groupes distincts, mais ces deux groupes ne correspondent pas aux deux classes. Les deux groupes correspondent aux deux sessions d'enregistrement dont le jeu de donnée est composé.

4.4.5 Résultats expérimentaux

Les performances de ces algorithmes seront évaluées sur une banque de données composée de 3 jeux de données *cue-based* et multi-classes :

1. **Compétition BCI IV - jeu IIa** : ce jeu de données a été fourni par le *Laboratory of Brain-Computer Interfaces (BCI-Lab)*, *Graz University of Technology* lors de la

compétition BCI IV³. Il est composé de 9 utilisateurs. Les signaux sont enregistrés sur 22 électrodes et comportent 576 réalisations au total pour 4 classes (144 réalisations par classe). Les classes correspondent à l'imagination d'un mouvement de la main droite, de la main gauche, du pied et de la langue. Pour chaque réalisation, on utilisera 2 secondes de données, une demi-seconde après la consigne.

2. **Compétition BCI III - jeu IIIa** : ce jeu de données, également fourni par le BCI-Lab de Graz, est dédié à la compétition BCI III⁴. Ce jeu de données comporte 3 utilisateurs. Les signaux sont enregistrés sur 60 électrodes et le jeu de données comporte 360 réalisations au total pour 4 classes (90 réalisations par classe). Ici aussi, les classes correspondent à l'imagination d'un mouvement de la main droite, de la main gauche, du pied et de la langue. Pour chaque réalisation, on utilisera également 2 secondes de données, une demi-seconde après la consigne. La principale différence entre ce jeu de données et le précédent est le nombre d'électrodes.
3. **Laboratoire** : ce jeu de données a été enregistré dans notre laboratoire au cours de la première campagne de test Brain-Switch. Il est composé de 8 sujets, les signaux étant enregistrés sur 16 électrodes. Il comporte un total de 60 réalisations sur 3 classes (20 par classe). Les 3 classes sont le mouvement imaginé de la main droite, le mouvement imaginé du pied et également une tâche repos où l'utilisateur ne fait rien. Pour chaque réalisation, on utilisera les 3 secondes de données consécutives à la présentation de la consigne.

Les deux premiers jeux de données sont enregistrés dans des conditions bien maîtrisées, les signaux sont donc relativement propres. Le premier, comportant 22 électrodes et 576 réalisations, est idéal pour évaluer la performance de nos algorithmes, la dimension de la variété ($m = 253$) étant inférieure au nombre de réalisations. Le second qui comporte 60 électrodes, permet de se confronter aux problèmes liés à la très grande dimension de la variété ($m = 1830$). Enfin, le troisième jeu de données est enregistré au laboratoire dans des conditions de vie courante i.e. dans un bureau de travail standard équipé de plusieurs ordinateurs et autres dispositifs électriques sources de bruit. Il comporte également peu de réalisations ($I = 60$) pour une variété de dimension $m = 136$. On est donc en présence d'un cas difficile avec des signaux bruités et un faible nombre de réalisations.

Les performances seront données en pourcentage de réalisations correctement classées. Le taux de classification correcte réalisée par chance est de 25% pour les deux jeux de données 4 classes et de 33% pour le jeu de données 3 classes. Les performances seront comparées à celles obtenues par une implémentation de l'algorithme de référence constitué d'un filtrage spatial par CSP **multi-classe** (décrit section 3.2.4) et d'une classification par classifieur de Bayes⁵ (fonction *classify* de Matlab) sur la log-variance des signaux filtrés. Les 8 meilleurs filtres spatiaux seront sélectionnés par la méthode basée sur l'information mutuelle [50] (section 3.2.4, équation (3.14)).

k -NN

La figure 4.10 donne l'évolution des performances moyennes en classification pour l'algorithme k -NN en fonction du nombre de voisins (en pourcentage du nombre I de réalisations disponible dans le jeu de données). Comme on s'y attend, les performances pour la distance Euclidienne sont faibles. Les 3 autres distances offrent des performances proches, avec un maximum de 63% pour un nombre de voisins compris dans une plage de 10 à 25% du nombre total de réalisations. La distance Riemannienne offre les meilleures performances, suivie par la distance de Kullback-leibler puis par la distance Log-Euclidienne.

3. <http://www.bbc.de/competition/iv/>

4. <http://www.bbc.de/competition/iii/>

5. Les jeux de données étant multi-classe, nous n'utiliserons pas la LDA.

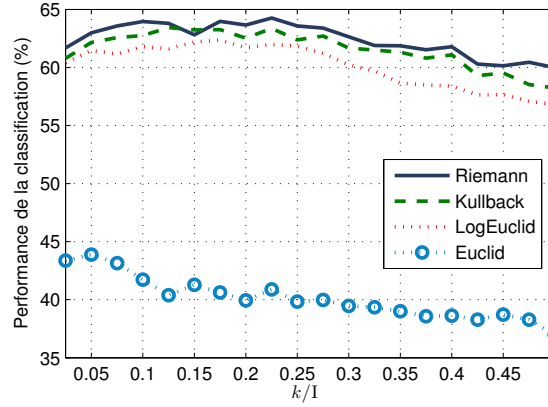


FIGURE 4.10 – Performances en classification pour l'algorithme k -NN en fonction du rapport k/I i.e du nombre de voisins sur le nombre total de points disponibles dans la base d'apprentissage. Validation croisée (30-fold).

La table 4.1 donne les performances détaillées pour les différents jeux de données et pour un nombre $k = 15$ de voisins. Utilisé avec la distance Riemannienne, l'algorithme k -NN réalise 64% de classification correcte. En comparaison, l'algorithme de référence réalise un score de 65.5%. Sur la plupart des sujets, la méthode de référence offre des résultats légèrement meilleurs. Pour certains sujets (A8, A9, k6, UB, UI) l'algorithme k -NN apporte une amélioration significative (+10%). L'utilisation de la distance Euclidienne se montre inefficace dans la plupart des cas.

MDM

La table 4.2 présente les résultats pour la méthode MDM. On observe des performances en légère amélioration par rapport à la méthode de référence avec 66% de classification correcte pour l'utilisation de la distance Riemannienne. De la même manière que pour l'algorithme k -NN, on retrouve les mêmes tendances, avec la distance Euclidienne inefficace, les 3 autres distances offrant des résultats similaires.

Sur des données propres (jeu IV-IIa), la distance Riemannienne offre des résultats équivalents à la méthode de référence. En revanche, cette méthode résiste bien à l'augmentation de la dimension de l'espace, avec des performances moyennes en légère augmentation sur les jeux de données III-IIIa et Labo.

Conclusion

L'adaptation des algorithmes de l'état de l'art à une nouvelle définition de la distance est donc efficace pour résoudre notre problématique de classification. Cette approche, bien que n'offrant pas une grande amélioration en termes de performances, est extrêmement simple à mettre en œuvre et constitue une première tentative très encourageante d'utilisation de la géométrie Riemannienne appliquée aux ICMs. Les résultats obtenus avec cette nouvelle approche, permettant de classer directement les matrices de covariance, sont pénalisés par l'emploi d'algorithmes de classification qui ne permettent pas de s'affranchir des informations non-pertinentes contenues dans les signaux. Ainsi, on peut s'attendre à une augmentation conséquente des performances avec l'emploi d'algorithmes de classification plus évolués. L'algorithme MDM a été présenté lors de la conférence LVA/ICA 2010 [8], ainsi que dans un article en cours de publication dans le journal *IEEE Transaction on Biomedical Engineering* [10].

		<i>k</i> -NN - Type de Distance				
	User	Riemann	Kullback	LogEuclidienne	Euclidienne	CSP
IV-IIa	A1	74.2	72.3	76.3	49.3	77.8
	A2	44.4	45.8	44.4	29.7	44.4
	A3	80.4	80.9	78.7	45.3	82.2
	A4	52.2	53.4	50.9	37.9	58.0
	A5	37.3	35.2	38.5	28.3	41.0
	A6	44.3	43.5	41.5	32.6	49.2
	A7	73.0	72.7	70.2	33.5	80.6
	A8	80.9	81.6	79.8	52.0	71.2
	A9	84.1	83.2	80.8	54.2	79.4
	moyenne	63.5	63.2	62.3	40.3	64.8
III-IIIa	k3	86.4	83.3	84.7	52.8	91.1
	k6	53.3	47.5	49.6	38.3	44.2
	l1	67.9	63.3	57.1	39.6	81.7
	moyenne	69.2	64.7	63.8	43.6	72.3
Laboratoire	UA	61.7	61.7	61.7	46.7	75.0
	UB	70.0	71.7	70.0	40.0	61.7
	UE	56.7	51.7	53.3	51.7	63.3
	UG	46.7	45.0	50.0	26.7	51.7
	UH	45.0	41.7	36.7	43.3	45.0
	UI	88.3	86.7	86.7	65.0	81.7
	UJ	75.0	76.7	75.0	53.3	68.3
	UL	60.0	58.3	58.3	30.0	63.3
	moyenne	62.9	61.7	61.5	44.6	63.8
	Moyenne	64.1	62.8	62.2	42.5	65.5
	Std	16	16.6	16.3	10.5	15.6

TABLE 4.1 – Performances pour la méthode de classification des k plus proches voisins. Utilisation de 15 voisins ($k = 15$) en validation croisée (*30-fold*).

		MDM - Type de Distance				
	User	Riemann	Kullback	LogEuclidienne	Euclidienne	CSP
IV-IIa	U1	78.9	78.0	77.8	40.1	77.8
	U2	48.3	46.6	44.7	25.9	44.4
	U3	77.5	76.0	72.7	41.7	82.2
	U4	58.3	57.3	54.5	37.6	58.0
	U5	44.2	42.8	40.9	23.0	41.0
	U6	47.0	47.3	44.9	33.8	49.2
	U7	74.1	73.4	70.1	33.3	80.6
	U8	75.7	75.5	72.4	37.2	71.2
	U9	76.4	75.9	74.7	48.8	79.4
	moyenne	64.5	63.6	61.41	35.7	64.8
III-IIIa	k3	87.2	84.7	83.9	41.1	91.1
	k6	56.7	55.8	53.3	39.2	44.2
	l1	80.0	79.2	70.0	29.2	81.7
	moyenne	74.6	73.2	69.1	36.5	72.3
Laboratoire	UA	75.0	73.3	73.3	41.7	75.0
	UB	66.7	66.7	73.3	41.7	61.7
	UE	58.3	55.0	55.0	40.0	63.3
	UG	55.0	55.0	48.3	25.0	51.7
	UH	51.7	51.7	50.0	40.0	45.0
	UI	81.7	83.3	81.7	66.7	81.7
	UJ	75.0	76.7	73.3	53.3	68.3
	UL	60.0	60.0	66.7	25.0	63.3
	moyenne	65.4	65.2	65.2	41.7	63.8
	moyenne	66.4	65.7	64.1	38.2	65.5
	std	13.3	13.4	13.6	10.4	15.6

TABLE 4.2 – Performances pour la méthode de classification MDM en validation croisée (30-fold).

Géométrie Riemannienne :

Espace tangent

Le chapitre précédent a présenté des techniques de classification dans la variété Riemannienne. Ces méthodes permettent une mise en œuvre simple de la classification de signaux EEG, en les appliquant directement sur les matrices de covariance spatiale extraites sur des portions de signaux EEG bruts ou préalablement filtrés fréquemment. Dans ce contexte, la problématique principale vient du fait que les signaux EEG sont extrêmement bruités, et donc que les caractéristiques (matrices de covariance) sur lesquelles on applique la classification perdent en spécificité. En s'appuyant sur le concept de plan tangent, nous présenterons dans ce chapitre des méthodes permettant de résoudre cette problématique.

5.1 Introduction

Il est généralement d'usage de filtrer les caractéristiques pour s'affranchir d'une partie du bruit qui perturbe le calcul des distances. Une manière d'effectuer ce filtrage est de séparer l'espace des caractéristiques en deux sous-espaces, un espace signal et un espace bruit, et de classer les données uniquement dans le sous-espace signal. On peut alors utiliser des techniques de séparation de sources, une analyse en composantes principales (PCA) ou une analyse discriminante linéaire (LDA).

Ces outils sont généralement conçus pour des espaces euclidiens et utilisent des techniques de projection pour séparer ces sous-espaces. Ces techniques ne sont pas directement utilisables dans la variété Riemannienne. Il est alors nécessaire de définir des nouveaux outils afin de mener à bien ces opérations directement sur les matrices de covariance dans le contexte de la géométrie Riemannienne.

5.1.1 Opérateur exponentiel et logarithmique

Pour chaque point $\Sigma \in P(n)$ de la variété, il existe un espace tangent composé de l'ensemble des vecteurs tangents à Σ . Chaque vecteur tangent \mathbf{S}_i peut être interprété comme la dérivée à $t = 0$ de la géodésique $\Gamma_i(t)$ reliant le point Σ à un point Σ_i particulier de la variété :

$$\mathbf{S}_i = \left. \frac{d\Gamma_i(t)}{dt} \right|_{t=0} = \Sigma^{\frac{1}{2}} \log \left(\Sigma^{-\frac{1}{2}} \Sigma_i \Sigma^{-\frac{1}{2}} \right) \Sigma^{\frac{1}{2}}. \quad (5.1)$$

On nomme alors projection logarithmique l'opérateur défini par :

$$\text{Log}_{\Sigma}(\Sigma_i) = \mathbf{S}_i = \Sigma^{\frac{1}{2}} \log \left(\Sigma^{-\frac{1}{2}} \Sigma_i \Sigma^{-\frac{1}{2}} \right) \Sigma^{\frac{1}{2}} \quad (5.2)$$

La projection exponentielle $\Sigma_i = \text{Exp}_{\Sigma}(\mathbf{S}_i)$, définie par

$$\text{Exp}_{\Sigma}(\mathbf{S}_i) = \Sigma_i = \Sigma^{\frac{1}{2}} \exp \left(\Sigma^{-\frac{1}{2}} \mathbf{S}_i \Sigma^{-\frac{1}{2}} \right) \Sigma^{\frac{1}{2}} \quad (5.3)$$

est l'opération inverse. Ici les opérateurs $\exp(\cdot)$ et $\log(\cdot)$ sont les opérateurs exponentiel et logarithmique d'une **matrice**.

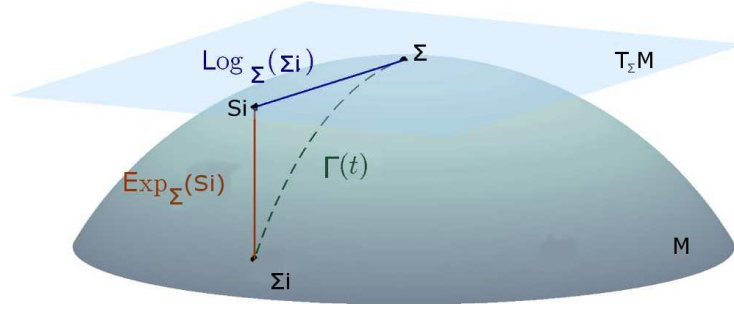


FIGURE 5.1 – Illustration du plan tangent au point Σ , avec \mathbf{S}_i le vecteur tangent de Σ_i et $\Gamma_i(t)$ la géodésique entre Σ et Σ_i . Le passage dans le plan tangent s'effectue par l'intermédiaire de la carte logarithmique et le retour dans la variété par la carte exponentielle.

Ces deux opérateurs permettent donc le passage de la variété Riemannienne vers le plan tangent et vice-versa. Cette procédure est présentée géométriquement par la Fig. 5.1.

On peut définir également la distance Riemannienne à partir de ce vecteur tangent :

$$\begin{aligned} \delta_R(\Sigma, \Sigma_i) &= \|\text{Log}_\Sigma(\Sigma_i)\|_\Sigma = \|\mathbf{S}_i\|_\Sigma \\ &= \|\Sigma^{-\frac{1}{2}} \text{Log}_\Sigma(\Sigma_i) \Sigma^{-\frac{1}{2}}\|_{\mathbf{I}_C} = \|\Sigma^{-\frac{1}{2}} \text{Log}_\Sigma(\Sigma_i) \Sigma^{-\frac{1}{2}}\|_F \end{aligned} \quad (5.4)$$

$$= \|\text{upper}\left(\Sigma^{-\frac{1}{2}} \text{Log}_\Sigma(\Sigma_i) \Sigma^{-\frac{1}{2}}\right)\|_2 = \|\mathbf{s}_i\|_2. \quad (5.5)$$

avec l'opérateur $\text{upper}(\cdot)$ qui consiste à garder la partie triangulaire supérieure de la matrice symétrique et de la vectoriser en appliquant un poids unitaire pour ses éléments diagonaux et un poids de $\sqrt{2}$ pour les éléments hors diagonale [116]. Ici $\mathbf{s}_i = \text{upper}\left(\Sigma^{-\frac{1}{2}} \text{Log}_\Sigma(\Sigma_i) \Sigma^{-\frac{1}{2}}\right)$ est le vecteur de dimension m de l'espace tangent normalisé.

Sous certaines conditions sur Σ et Σ_i [116], l'équation (5.4) conduit à une approximation en termes de distance entre les éléments de l'espace tangent et ceux de la variété, c'est-à-dire :

$$\forall i, j \quad \delta_R(\Sigma_i, \Sigma_j) \approx \|\mathbf{s}_i - \mathbf{s}_j\|_2. \quad (5.6)$$

Pour que les conditions menant à cette approximation soient vérifiées, il faut que les Σ_i soient distribués localement sur la variété, c'est-à-dire localisés sur une petite partie de la variété et que Σ , le point où est calculé l'espace tangent, soit la moyenne des Σ_i .

Ce résultat est fondamental, il nous permet **de travailler dans le plan tangent avec des outils de la géométrie Euclidienne en conservant une bonne représentation de la variété Riemannienne**.

5.2 Filtrage géodésique

5.2.1 Principe

A partir de ces nouveaux outils, nous allons chercher à appliquer des opérations dans le but de dé-bruiter les données et ainsi augmenter les performances de la classification. La procédure de filtrage se déroule en 3 étapes

1. **Projection dans le plan tangent** : on projette l'ensemble des données dans le plan tangent grâce à l'opérateur logarithmique, équation (5.2). On obtient un nouvel ensemble de matrices symétriques \mathbf{S}_i que l'on vectorise et pondère pour obtenir des vecteurs \mathbf{s}_i de dimension m .

2. **Filtrage dans le plan tangent** : on effectue les opérations de filtrage classique sur ces nouvelles caractéristiques \mathbf{s}_i .
3. **Rétro-projection dans la variété** : après filtrage, on renvoie les vecteurs dans la variété Riemannienne par l'intermédiaire de l'opérateur exponentiel, équation (5.3). On peut ensuite appliquer une des techniques de classification dans la variété Riemannienne présentées dans la section 4.4.

5.2.2 Filtrage par projection linéaire

Soit $\Omega = \{\Sigma_i\}$ un ensemble de matrices de covariance qu'on souhaite "filtrer". On estime en premier lieu la matrice moyenne de l'ensemble des données $\bar{\Sigma} = \mathfrak{G}(\Sigma_i)$. Cette matrice moyenne sert de point de référence pour le calcul de l'espace tangent. En effet, comme vu à la section 5.1.1, c'est au niveau de la moyenne géométrique que l'on obtient la meilleure approximation sur les distances entre les points entre le plan tangent et la variété. On projette ensuite les données pour obtenir les vecteurs de l'espace tangent réduit :

$$\mathbf{s}_i = \text{upper} \left(\bar{\Sigma}^{-\frac{1}{2}} \text{Log}_{\bar{\Sigma}}(\Sigma_i) \bar{\Sigma}^{-\frac{1}{2}} \right) \quad (5.7)$$

avec l'opérateur $\text{upper}(\cdot)$ défini section 5.1.1. Les vecteurs ainsi obtenus sont de dimension m .

Nous allons appliquer ensuite un filtrage de type linéaire, c'est-à-dire que l'opération de filtrage prendra la forme suivante :

$$\tilde{\mathbf{s}}_i = \mathbf{W} \mathbf{s}_i \quad (5.8)$$

avec $\mathbf{W} = [\mathbf{w}_1 \dots \mathbf{w}_J]^T \in \mathbb{R}^{J \times m}$ une matrice constituée par la concaténation des $J < m$ filtres de m coefficients. Les filtres \mathbf{w}_j peuvent être obtenus indifféremment par plusieurs méthodes, en fonction des caractéristiques du filtrage que l'on souhaite réaliser.

Le filtrage réalisé par l'équation (5.8) a pour effet de réduire la dimension de l'espace de représentation, en plus d'effectuer un changement de base, ce qui empêche en retour d'utiliser la carte exponentielle pour renvoyer les données dans la variété. Pour palier à ce problème on utilisera comme projecteur la matrice $\hat{\mathbf{W}} \in \mathbb{R}^{m \times m}$ qui renvoie les données filtrées dans leur base originelle. Ce projecteur est donné par l'équation suivante :

$$\hat{\mathbf{W}} = \mathbf{W}^T (\mathbf{W} \mathbf{W}^T)^{-1} \mathbf{W} \quad (5.9)$$

alors

$$\hat{\mathbf{s}}_i = \hat{\mathbf{W}} \mathbf{s}_i \quad (5.10)$$

Finalement, on utilise l'opérateur $\text{upper}^{-1}(\cdot)$ pour remettre le vecteur sous forme de matrice symétrique puis on utilise l'opérateur exponentiel pour renvoyer l'élément filtré dans la variété Riemannienne :

$$\hat{\Sigma}_i = \text{Exp}_{\bar{\Sigma}} \left(\bar{\Sigma}^{-\frac{1}{2}} \text{upper}^{-1}(\hat{\mathbf{s}}_i) \bar{\Sigma}^{-\frac{1}{2}} \right) \quad (5.11)$$

Chaque filtre \mathbf{w}_j pointe une direction de l'espace tangent. L'opération effectuée par le filtrage équation (5.10) projette les données sur un hyperplan défini par J droites de l'espace tangent. Le passage dans la variété projette alors les données sur une sous-variété définie par J géodésiques. On appelle ainsi cette opération filtrage géodésique.

Une approche similaire a été développée dans [39] sous le nom de *Principal Geodesic Analysis*. Il s'agissait d'effectuer une analyse en composantes principales dans le plan tangent et de reconstruire les formes principales à partir des modes de variations de la PCA dans le cadre de tenseur de diffusion en imagerie médicale.

Analyse en géodésiques discriminante de Fisher (FGDA)

Notre approche est différente de [39] car nous allons utiliser cette technique de filtrage dans le but de dé-bruiter les matrices de covariance dans l'optique d'augmenter la séparabilité des classes dans la variété Riemannienne.

Les filtres seront ainsi calculés par une analyse discriminante de Fisher, et en particulier sa généralisation au cas multi-classe, décrite dans la section 3.3.2. Étant donné la grande dimension de l'espace tangent, on utilisera une estimation régularisée de la matrice de covariance intra-classe des données. Cette estimation est réalisée par la technique de *shrinkage* (équation B.8), dont le paramètre de shrinkage est calculé automatiquement selon la méthode décrite en Annexe A et présentée dans [18].

La matrice de covariance inter-classes \mathbf{S}_b étant de rang $K - 1$, on utilisera généralement moins de $K - 1$ filtres pour la projection. On applique ensuite la procédure de filtrage décrite dans la section 5.2.1.

La figure 5.2 illustre ce principe sur des données synthétiques à deux classes et pour des matrices 2×2 . Les données sont projetées sur le plan tangent, puis les composantes qui maximisent la séparabilité des classes sont calculées et les données filtrées selon ces directions. Les données filtrées, qui appartiennent à un sous-espace du plan tangent, sont rétro-projetées dans la variété.

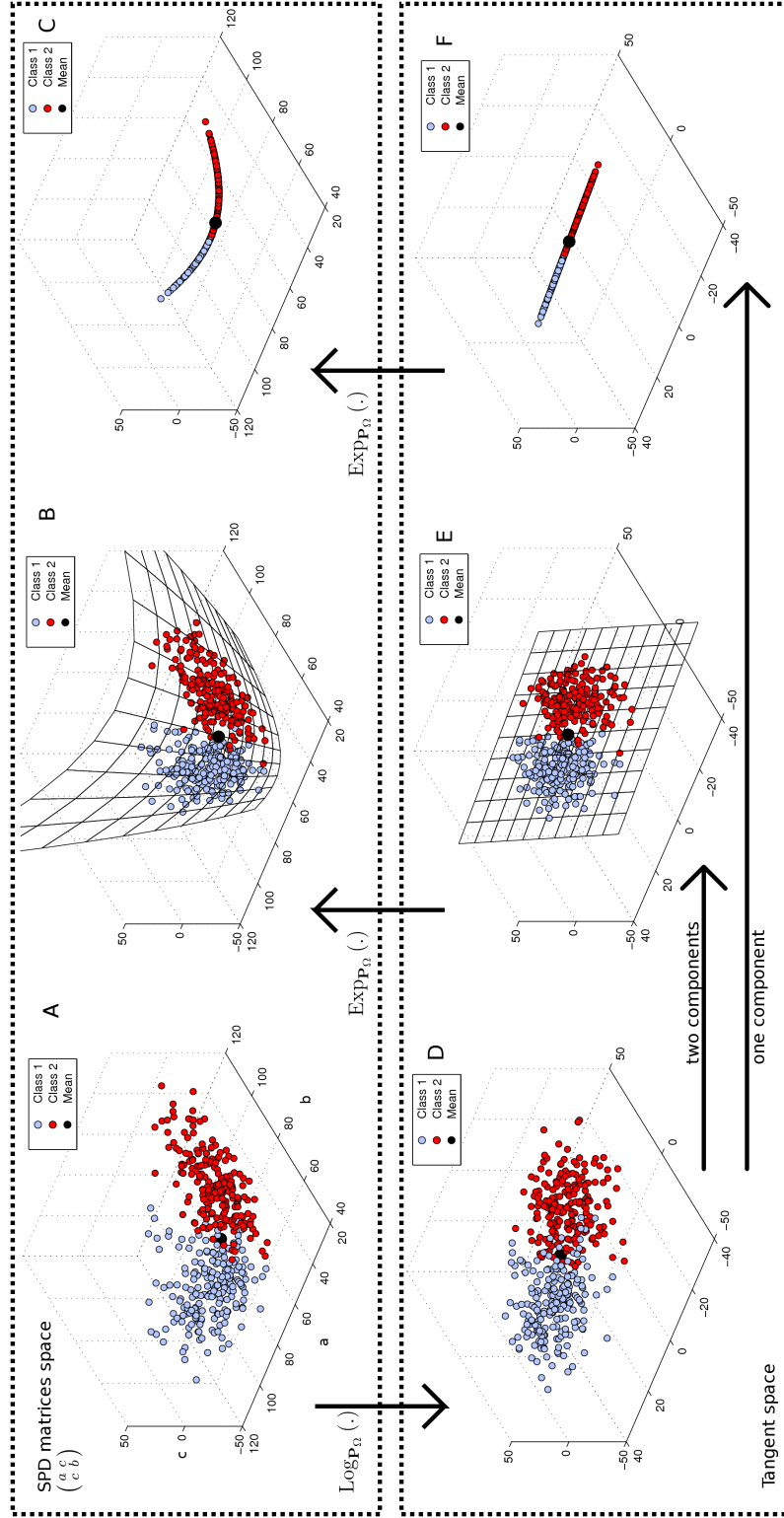


FIGURE 5.2 – Filtrage géodésique suivant un ou deux filtres calculés par analyse discriminante de Fisher. Les données filtrées dans l'espace tangent se retrouvent sur une sous-variété après rétro-projection dans l'espace de départ.

5.2.3 Résultats

La table 5.1 donne les résultats du filtrage géodésique utilisé en combinaison des deux algorithmes de classification supervisée dans la variété (avec distance Riemannienne) présentés dans la section 4.4.

		sans filtrage		avec filtrage		
	User	k -NN	MDM	k -NN	MDM	CSP
IV-IIa	A1	74.2	78.9	80.8	79.5	77.8
	A2	44.4	48.3	48.0	50.1	44.4
	A3	80.4	77.5	88.8	88.6	82.2
	A4	52.2	58.3	61.5	62.5	58
	A5	37.3	44.2	43.6	46	41
	A6	44.3	47.0	54.8	57.7	49.2
	A7	73.0	74.1	80.6	82.2	80.6
	A8	80.9	75.7	85.2	84.5	71.2
	A9	84.1	76.4	87.5	88.2	79.4
	moyenne	63.5	64.5	70.1	71	64.8
III-IIIa	k3	86.4	87.2	94.7	94.4	91.1
	k6	53.3	56.7	66.3	63.8	44.2
	l1	67.9	80.0	81.7	82.5	81.7
	moyenne	69.2	74.6	80.9	80.3	72.3
Laboratoire	UA	61.7	75.0	76.7	73.3	75.0
	UB	70	66.7	68.3	70	61.7
	UE	56.7	58.3	83.3	83.3	63.3
	UG	46.7	55	50	56.7	51.7
	UH	45.0	51.7	48.3	51.7	45
	UI	88.3	81.7	83.3	85	81.7
	UJ	75	75	80	78.3	68.3
	UL	60	60	61.7	65	63.3
	moyenne	62.9	65.4	69	70.4	63.8
	Moyenne	64.1	66.4	71.3	72.1	65.5
	Std	16	13.3	15.8	14.6	15.6

TABLE 5.1 – Performances pour les méthodes de classification k -NN et MDM en combinaison du filtrage géodésique par analyse discriminante en validation croisée (30-*fold*)

L'application d'un filtrage dans l'espace tangent, en combinaison avec une classification dans la variété, permet d'augmenter de manière significative les performances pour tous les utilisateurs. Ainsi avec une classification de type MDM, on obtient un taux de classification correcte de 72%, ce qui constitue une amélioration de 7% par rapport à la méthode de référence. L'application d'une analyse statistique par l'intermédiaire d'un t-test par paires pour l'hypothèse $\mathcal{H}_0 : \mu_1 > \mu_2$ (les performances moyennes de la méthode 1 sont meilleures que celles de la méthode 2), dont les p-valeurs sont données table 5.2, montrent que cette amélioration par rapport à la méthode de référence est très significative ($p < 0.05$).

	IV-IIa	III-IIIa	Labo	Combinaison
FG+kNN vs. CSP	0.0025	0.168	0.044	0.0016
FG+MDM vs. CSP	0.0005	0.156	0.0118	0.0001

TABLE 5.2 – p-valeurs pour un t-test par paires sous l'hypothèse $\mathcal{H}_0 : \mu_1 > \mu_2$. La combinaison sur les différents tests est effectuée par la méthode de Edgington [32].

L'utilisation d'une estimation régularisée de la matrice de covariance \mathbf{S}_w est particulièrement efficace. En effet, les résultats sur le jeu de données III-IIIa ne montrent pas de dégradation dans les performances. On note même une augmentation de plus de 20% sur le sujet k6 par rapport à la méthode de référence. Enfin, on observe également de bonnes performances sur le jeu de données enregistré au laboratoire, avec 3 sujets (UA,UE,UJ) montrant une amélioration supérieure à 10%. Ce dernier jeu étant fortement bruité, il était donc prévisible d'obtenir de tels résultats.

Il est intéressant d'observer que l'introduction du filtrage géodésique diminue l'écart de performance entre la méthode k -NN et MDM. On rappellera également que cette méthode ne nécessite le réglage d'aucun paramètre, le nombre de filtres utilisés étant conditionné par le nombre K de classes. Ainsi, cette méthode offre d'excellents résultats pour un minimum d'efforts, elle constitue donc une solution réellement intéressante pour la mise en œuvre rapide d'ICMs.

Cette procédure de filtrage géodésique a été présentée à la conférence LVA/ICA en 2010 [8].

5.3 Classification dans le plan tangent

La section précédente a présenté l'utilisation du plan tangent dans le but de filtrer les matrices de covariance avant classification dans la variété Riemannienne. Si dans certaines applications, le retour dans la variété Riemannienne peut être intéressant, il est finalement équivalent de rester dans le plan tangent pour effectuer la classification.

5.3.1 Introduction

Le passage dans l'espace tangent peut être considéré comme une étape de pré-traitement, ou d'extraction de caractéristiques, permettant de se placer dans un espace où l'information spatiale contenue dans les matrices de covariance est directement accessible par des classifieurs linéaires ou non-linéaires. Ainsi, on pourra utiliser des classifieurs plus évolués, tel que la LDA, les SVM ou la régression logistique. On pourra observer en figure 5.3 la distribution des données pour une des variables du plan tangent et la comparer à celle obtenue après un filtrage spatial par CSP.

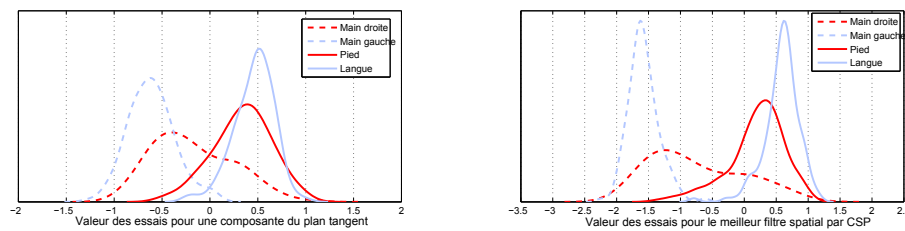


FIGURE 5.3 – Distribution des données sur une caractéristique du plan tangent (à gauche) et sur une caractéristique obtenue par filtrage spatial par CSP (à droite). Sujet A1.

Bien que la variable obtenue par CSP soit plus discriminante, les distributions sont relativement semblables. Le passage dans le plan tangent permet donc d'avoir accès à des caractéristiques d'une qualité similaire à celles obtenue par CSP. De plus on notera que le passage dans le plan tangent est une opération non-supervisée, contrairement à la CSP, ce qui évite donc les problématiques liées au sur-apprentissage de la méthode d'extraction de caractéristiques.

On utilisera également la moyenne géométrique de l'ensemble des données comme point de passage dans le plan tangent, ce qui permet de **minimiser l'erreur entre la distance**

Euclidienne dans le plan tangent et la distance Riemannienne dans la variété. Une vérification sur nos données EEG montre que cette approximation est de l'ordre de 2% c'est-à-dire qu'en moyenne, on observe un écart de 2% entre la valeur de la distance Riemannienne entre deux points de la variété et la distance Euclidienne dans le plan tangent entre les deux vecteurs tangents correspondants. Cette approximation est suffisamment faible pour être négligeable.

La problématique principale induite par le passage dans le plan tangent est la grande dimension de cet espace. Pour C électrodes, la dimension est $m = \frac{C(C+1)}{2}$. En considérant les montages classiques à 16, 32 ou 64 électrodes, le plan tangent sera respectivement de dimension 135, 528 et 2080. On arrive donc rapidement à avoir la dimension de l'espace qui dépasse le nombre de réalisations utilisées pour entraîner le classifieur. En pratique, cela entraînera, dans la majorité des cas, du sur-apprentissage de la part de l'algorithme de classification, et le classifieur obtenu ne sera plus généralisable. Ce phénomène est connu sous le nom de fléau de la dimension (*Curse of dimensionality*).

Pour résoudre ce problème, on dispose globalement de 3 approches que l'on peut appliquer seules ou combinées :

1. **Sélection d'électrodes.** Ce point particulier fera l'objet du chapitre 6.
2. **Sélection de caractéristiques.**
3. **Classification régularisée.**

5.3.2 Sélection de caractéristiques et classification

On peut faire l'hypothèse que l'ensemble des m dimensions de l'espace ne sont pas toutes utiles pour la classification et ne portent pas toute une information complémentaire. Ainsi on pourra sélectionner un sous-ensemble de caractéristiques dans le but de réduire la taille de l'espace avant d'y effectuer la classification. Les variables de l'espace tangent sont fortement corrélées entre elles. Lorsqu'on cherche à effectuer une sélection de variables en les considérant séparément, on risque alors de sélectionner successivement des variables portant la même information. Afin d'éviter ce problème, une première étape consiste à dé-corréler les caractéristiques, en effectuant par exemple une PCA. Soit $\mathbf{s}_i \in \mathbb{R}^{m \times 1}$ un vecteur du plan tangent, et $\mathbf{S} = [\mathbf{s}_1 \dots \mathbf{s}_I] \in \mathbb{R}^{m \times I}$ la matrice formée de ces I vecteurs, les m variables sont alors décorrélées par l'intermédiaire d'une décomposition en valeurs singulières (SVD) :

$$\mathbf{S} = \mathbf{U}\mathbf{\Lambda}\mathbf{V}^T \quad (5.12)$$

avec $\mathbf{U} \in \mathbb{R}^{m \times m}$ et $\mathbf{V} \in \mathbb{R}^{I \times I}$ deux matrices orthogonales et $\mathbf{\Lambda} \in \mathbb{R}^{m \times I}$ la matrice diagonale composée des valeurs singulières de \mathbf{S} .

On projette les vecteurs \mathbf{S} du plan tangent dans la base orthogonale \mathbf{U} :

$$\mathbf{S}_o = \mathbf{U}^T \mathbf{S} \quad (5.13)$$

pour obtenir un nouveau jeu de vecteurs \mathbf{S}_o où les m variables sont décorrélées (et de matrice de covariance $\mathbf{\Lambda}^2$).

On applique ensuite une analyse de variance (Anova) pour chacune des m dimensions de \mathbf{S}_o afin de rechercher les variables pour lesquelles les distributions de chaque classe sont significativement différentes. Puis, on les trie en fonction des p-valeurs ainsi obtenues. Enfin, on utilise un algorithme wFDR (weighted False Discovery Rate) [16] pour sélectionner le nombre optimal de variables. Les valeurs singulières de \mathbf{S} sont alors utilisées comme poids dans la procédure wFDR afin de donner la priorité aux variables qui représentent au mieux la structure des données.

Cette procédure nous permet d'obtenir un jeu de variables de dimension réduite sur lequel on appliquera un algorithme de classification standard tel que la LDA ou le SVM

dans le plan tangent. Dans notre cas, nous utiliserons, après sélection de ces caractéristiques, un classifieur de Bayes (fonction *classify* de Matlab), identique à celui utilisé par la méthode de référence.

5.3.3 Classification par régression logistique régularisée

La régression logistique régularisée, présentée section 3.3.4, effectue automatiquement la sélection des variables d'intérêt sur lesquelles la classification sera appliquée. On pourra alors utiliser directement cette méthode de classification dans l'espace tangent, avec peu de risque de sur-apprentissage.

La figure 5.4 montre la valeur de chaque coefficient des β_k (poids des variables explicatives) pour un problème à 4 classes. On observe ainsi que la majorité des coefficients est nulle.

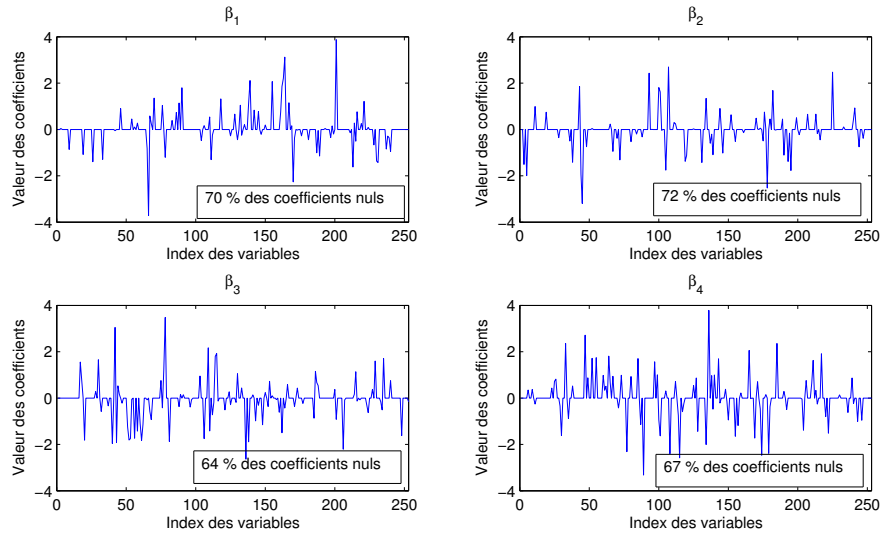


FIGURE 5.4 – Coefficients des 4 vecteurs β_k pour la régression logistique régularisée. User A1, $\alpha = 0.95$

On pourra également augmenter la puissance de la régularisation en réduisant une première fois l'espace tangent par une ACP régularisée (diagonalisation de la matrice de covariance estimée par *shrinkage*) :

$$\mathbf{\Lambda} = \mathbf{U}^T \mathbf{S}_w \mathbf{U} \quad (5.14)$$

où $\mathbf{\Lambda} = \text{diag}(\{\lambda_n\})$ est une matrice diagonale et \mathbf{S}_w est la matrice de covariance des caractéristiques \mathbf{S} . On conservera alors les m^* nouvelles composantes qui portent 80% de l'énergie des données :

$$m^* = \underset{l}{\operatorname{argmin}} \left| \frac{\sum_{n=1}^l \lambda_n}{\sum_{n=1}^m \lambda_n} - 0.8 \right| \quad (5.15)$$

Cette étape permet de réduire une première fois la dimension de l'espace en supprimant les composantes inutiles. La valeur de 80% est choisie arbitrairement. La régression logistique embarquant une sélection de variables, le choix de cette valeur n'est pas critique. En pratique cette méthode diminue de moitié la dimension de l'espace.

5.3.4 Kernel-SVM

De manière intéressante, le passage dans l'espace tangent peut être vu comme la projection des données (les matrices de covariance) par une fonction noyau qui aurait pour effet de linéariser l'espace de départ. Ainsi si les données dans la variété Riemannienne ont peu de chance d'être séparables par un hyperplan du fait de la courbure de cette variété, il est plus probable qu'elles le soient dans le plan tangent. Cette vision est proche de celle développée dans les SVM, on va donc chercher à expliciter la fonction noyau qui permettra de classer les matrices de covariance directement par un classifieur SVM.

Soit $\phi(\cdot)$ la fonction de projection, on définit généralement la fonction noyau $K(\cdot, \cdot)$ à partir du produit scalaire dans l'espace où les données ont été projetées. Soient Σ_i et Σ_j deux points de l'espace de départ, dans notre cas celui des matrices SPD, la fonction noyau $K(\cdot, \cdot)$ est définie par :

$$K(\Sigma_i, \Sigma_j) = \langle \phi(\Sigma_i), \phi(\Sigma_j) \rangle \quad (5.16)$$

Or, en géométrie Riemannienne, la métrique est justement définie à partir du produit scalaire dans le plan tangent, c'est-à-dire du produit scalaire dans l'espace projeté. Il est alors évident d'utiliser la métrique Riemannienne comme fonction noyau de notre SVM. A partir de l'équation de la métrique (4.14), de la fonction de passage dans le plan tangent (5.2) et de la définition de la fonction noyau (5.16), on obtient :

$$K_{\Sigma}(\Sigma_i, \Sigma_j) = \text{Tr} \left[\log \left(\Sigma^{-\frac{1}{2}} \Sigma_i \Sigma^{-\frac{1}{2}} \right) \log \left(\Sigma^{-\frac{1}{2}} \Sigma_j \Sigma^{-\frac{1}{2}} \right) \right] \quad (5.17)$$

avec Σ le point de passage dans le plan tangent, qui peut être considéré comme un paramètre du noyau.

On remarquera toutefois que l'on dispose d'une formulation explicite de la fonction de projection $\phi(\cdot)$ puisque c'est l'opérateur logarithmique donné en équation (5.2). On pourra alors, de manière équivalente, classer les matrices avec le noyau présenté ici, ou passer l'ensemble des données dans l'espace tangent et les classer en utilisant un classifieur SVM linéaire. Cependant, l'utilisation de la fonction noyau offre une approche élégante pour la classification des signaux EEG, car elle est extrêmement simple à mettre en œuvre. En particulier, si on considère la matrice identité comme point de passage dans l'espace tangent, la fonction noyau prend la forme suivante :

$$K_{I_C}(\Sigma_i, \Sigma_j) = \text{Tr} [\log(\Sigma_i) \log(\Sigma_j)] \quad (5.18)$$

ce qui sera certes moins efficace que l'utilisation du noyau (5.17) mais qui possède l'avantage de ne plus dépendre d'aucun paramètre (mis à part ceux requis par le SVM), et qui pourrait être appliqué de manière aveugle pour la classification des signaux EEG. L'implémentation de cette méthode revient alors simplement à ajouter une fonction noyau aux algorithmes SVM existants. De cette manière, on profite des avantages de la géométrie Riemannienne sans les inconvénients liés à l'implémentation relativement complexe de l'ensemble des méthodes. Finalement, la géométrie Riemannienne offre un moyen de choisir un noyau, ce qui généralement est une tâche ardue pour la classification SVM.

5.3.5 Résultats

La table 5.3 donne les résultats pour les 3 méthodes de classification dans le plan tangent et la table 5.4 donne les résultats en terme de p-valeurs pour les t-test appliqués sur ces résultats.

	User	Sélection	regLog	SVM	CSP
IV-IIa	A1	80.9	79.3	70.3	77.8
	A2	52.0	53.1	55.1	44.4
	A3	86.0	86.7	80.9	82.2
	A4	62.1	57.6	62.8	58.0
	A5	43.1	48.5	50.5	41.0
	A6	55.5	51	57.2	49.2
	A7	84.6	80	76.4	80.6
	A8	85.3	85.4	83.5	71.2
	A9	85.9	85.2	76.4	79.4
	moyenne	70.6	69.7	68.1	64.8
III-IIIa	k3	69.7	93.3	83.3	91.1
	k6	20.4	63.7	60.4	44.2
	l1	23.3	82.1	77.5	81.7
	moyenne	37.8	79.7	73.7	72.3
Laboratoire	UA	68.3	65	76.7	75.0
	UB	60.0	66.6	71.7	61.7
	UE	60.0	80	66.7	63.3
	UG	33.3	61.7	56.7	51.7
	UH	61.7	45	51.7	45.0
	UI	80.0	85	80.0	81.7
	UJ	73.3	75	78.3	68.3
	UL	48.3	53.3	63.3	63.3
	moyenne	60.6	66.5	68.1	63.8
	moyenne	61.7	69.9	69.0	65.5
	std	20.3	15.1	11.0	15.6

TABLE 5.3 – Performances pour les méthodes de classification dans l’espace tangent en validation croisée (30-*fold*)

	IV-IIa	III-IIIa	Labo	Combinaison
Selec vs. CSP	0.0007	0.95	0.77	0.66
regLog vs. CSP	0.009	0.175	0.218	0.01
SVM vs. CSP	0.11	0.43	0.012	0.028

TABLE 5.4 – p-valeurs pour un t-test par paires sous l’hypothèse $\mathcal{H}_0 : \mu_1 > \mu_2$. La combinaison sur les différents tests est effectuée par la méthode de Edgington [32].

Sélection : la méthode fondée sur la sélection de variables offre les meilleurs résultats sur le premier jeu de données, qui dispose d’un nombre suffisant de tests par rapport à la dimension de l’espace tangent, avec une amélioration significative ($p = 0.0007$). En revanche elle offre de très mauvaises performances dans le cas du jeu III-IIIa, la sélection de variables présentée en section 5.3.2 est donc peu efficace quand le nombre de dimensions est très important.

regLog : en revanche, la régression logistique régularisée traite très efficacement ce cas, elle offre les meilleures performances avec près de 80% de classification correcte sur le jeu de données III-IIIa et une amélioration de 7% presque significative ($p = 0.175$) sur ce jeu de données. Cette méthode montre également les meilleures performances globales, avec environ 70% de classification correcte sur l’ensemble des utilisateurs.

SVM : c'est avec la SVM qu'on obtient les meilleurs résultats sur le dernier jeu de données, avec une amélioration significative de 5% ($p = 0.012$). C'est également la méthode la plus simple puisqu'elle consiste uniquement à utiliser le noyau équation (5.17) (qui est équivalent à l'application d'un SVM linéaire dans l'espace tangent), sans sélection de caractéristiques.

On observe au final des performances légèrement inférieures au cas du filtrage géodésique. En revanche, on conserve une marge de progression importante, grâce à l'optimisation, spécifique à chaque utilisateur, des paramètres des différents classifieurs.

5.4 Conclusion

Ce chapitre a présenté plusieurs méthodes de classification originales fondées sur la géométrie Riemannienne de la variété des matrices de covariance. Ces nouvelles méthodes se classent sur 3 niveaux différents :

Classification dans la variété. À l'aide de la définition de la distance et de la moyenne, il est possible de classer directement dans la variété les matrices de covariance estimées sur des portions de signal EEG correspondant à des réalisations de diverses tâches mentales. La méthode offrant les meilleurs résultats est basée sur un algorithme de distance minimale aux centroïdes (MDM). Elle ne nécessite le réglage d'aucun paramètre et a donc une mise en œuvre très simple, tout en offrant des performances similaires aux algorithmes de l'état de l'art. De plus, elle se montre très efficace dans la plupart des cas, en particulier quand les données sont bruitées et peu nombreuses. Ce type de méthode est donc tout à fait indiqué pour la mise en œuvre rapide d'expérimentations EEG.

Filtrage géodésique et classification dans la variété. Grâce à l'utilisation de l'espace tangent, il est possible d'appliquer des opérations de "filtrage" sur les matrices de covariance et ainsi réduire le bruit contenu dans les données en vue d'augmenter les performances en classification des méthodes précédentes. Ainsi, un filtrage fondé sur une analyse discriminante linéaire a permis d'augmenter les performances de plus de 7% pour l'algorithme MDM. Les filtres doivent toutefois être calculés avec soin, souvent par des méthodes régularisées car le plan tangent est de grande dimension.

Classification dans le plan tangent. En projetant les données dans le plan tangent grâce à l'opérateur logarithmique, on pourra appliquer de manière équivalente des algorithmes de classification complexes, comme la SVM ou la régression logistique. L'utilisation des données dans le plan tangent ouvre la voie à de nombreuses améliorations et offre un nouvel espace de représentation des données EEG avec un accès direct à l'information spatiale sans utiliser de filtrage spatial. Enfin, il est possible d'explicitement une fonction noyau pour le SVM (équivalente à une classification SVM linéaire dans l'espace tangent) pour classer directement les matrices de covariance et ainsi réduire l'ensemble de la chaîne de traitement ICM à une unique opération.

Parmi les 4 distances présentées, la distance Riemannienne est la plus efficace. Cependant la distance de Kullback-Leibler et la distance Log-Euclidienne constitue une bonne alternative si l'on souhaite diminuer les temps de calcul, au prix d'une faible baisse (2%) des performances.

On dispose donc d'un ensemble complet de méthodes permettant de classer directement les matrices de covariance, et ainsi supprimer l'étape de filtrage spatial couramment utilisée en ICMs. La classification dans l'espace tangent offre des résultats très prometteurs. Des améliorations devront être faites dans l'optique de mieux sélectionner et réduire l'espace, ou encore pour ajuster automatiquement les paramètres des classifieurs.

Ces divers algorithmes ont fait l'objet d'une présentation à la conférence LVA/ICA en 2010 [8], ainsi qu'un article publié dans le journal *IEEE Transactions on Biomedical Engineering* [10].

Sélection d'électrodes

Nous avons abordé la problématique de sélection d'électrodes dès le début de la thèse. Cela nous est apparu comme un point clef dans la réalisation d'une ICM robuste et ergonomique. Ce chapitre propose deux méthodes originales de sélection de capteurs et de variables en général. La première est basée sur un critère de corrélation entre les variables et a fait l'objet d'une communication à la conférence GRETSI en 2009, la seconde exploite les concepts de la géométrie Riemannienne et a donné lieu à un dépôt de brevet ainsi qu'à une communication à la conférence Neural Engineering 2011.

6.1 État de l'art

But de la sélection d'électrodes. Comme il a été mentionné précédemment, lors de la réalisation d'une ICM, on associe une ou plusieurs tâches mentales à une ou plusieurs actions sur un effecteur. Chaque tâche mentale possède des caractéristiques fréquentielles, temporelles et spatiales qui lui sont propres et que l'on cherche à rehausser par des techniques de filtrage. Cependant, un choix judicieux de l'emplacement des électrodes peut permettre une réduction importante de leur nombre sans perte majeure d'information.

La mise en place d'une expérimentation EEG est **longue et fastidieuse**, bien que le matériel ait évolué sensiblement vers une simplification de la procédure, notamment grâce à l'apparition des électrodes actives (cf section 2.2.3). Cette étape reste néanmoins un frein important au développement des ICMs. En effet, la pose de chaque électrode nécessite un soin particulier en vue d'assurer un bon contact (faible impédance) avec le scalp, condition nécessaire à la minimisation du bruit et des artefacts. Si le domaine de la recherche peut se satisfaire de ces exigences, il n'en est pas de même pour une utilisation en vie courante. On utilise, dans la plupart des cas, un grand nombre d'électrodes dans le but de conserver un maximum d'information spatiale pouvant aider à la détection de la tâche mentale. Toutefois, les caractéristiques spatiales peuvent être très localisées, et plutôt que d'utiliser un grand nombre d'électrodes réparties sur l'ensemble du scalp, il est possible d'utiliser un nombre réduit d'électrodes positionnées correctement pour rehausser cette information spatiale. De plus, certains types de traitements montrent des limitations quand le nombre d'électrodes augmente, soit pour des raisons de sur-apprentissage pour les algorithmes de classification comme la LDA, soit pour des raisons calculatoires comme pour les méthodes Riemanniennes. Sélectionner intelligemment les électrodes a donc plusieurs avantages :

1. **Réduire le temps de pose des électrodes** et donc le temps de préparation avant l'utilisation de l'ICM,
2. **Diminuer la gêne** de l'utilisateur,
3. **Réduire le coût calculatoire** des traitements,
4. **Diminuer le coût** de l'équipement

Types de sélection. En ICM, on distingue principalement 2 types de sélection d'électrodes [109] :

1. La sélection spécifique à l'utilisateur (*patient specific*) construit un sous-ensemble d'électrodes dédié à chaque utilisateur. Elle a principalement pour but une augmentation des performances de l'ICM. On effectue d'abord une première session où l'on utilise un maximum d'électrodes afin de fournir un jeu de données sur lequel la procédure de sélection sera appliquée. On peut ensuite conduire les autres sessions avec un sous-ensemble réduit d'électrodes spécifique à l'utilisateur.
2. La sélection spécifique à l'application (*application specific*) qui cherche un sous-ensemble d'électrodes commun à tous les sujets et qui serait construit en fonction de l'application que l'on souhaite faire de l'ICM. Cette problématique a été abordée plus récemment [109]. On effectue d'abord une première session sur plusieurs sujets, on détermine le sous-ensemble optimal pour l'application donnée. On peut ensuite utiliser le sous-ensemble pour lancer une campagne de test sur un plus grand nombre de sujets.

Le principe de fonctionnement des algorithmes de sélection d'électrodes (ou de capteurs en général) est le suivant :

1. Premièrement, on définit un critère qui quantifie l'information utile portée par chaque électrode ou sous-ensemble d'électrodes. Les critères couramment utilisés sont : le rapport signal sur bruit [23] (nécessite la définition de ce qu'est le signal, tâche non triviale en BCI), l'information mutuelle [68] ou encore les coefficients des filtres spatiaux [125, 78].
2. Une fois le critère établi, on construit un sous-ensemble d'électrodes en cherchant à **maximiser** le critère en question.

Recherche du sous-ensemble : Dans certains cas, le critère choisi permet d'obtenir directement le sous-ensemble d'électrodes. En revanche, dans la plupart des cas, la construction du sous-ensemble se fait de manière **itérative**. Ceci est fait dans le cadre de procédure de recherches de sous-ensemble optimum (Quel est le sous-ensemble de C_1 électrodes parmi C maximisant le critère donné). Étant donné le nombre très important de combinaisons à tester il est parfois impossible de faire une recherche exhaustive. Cette problématique est proche de la sélection de variables explicatives en régression. Il est ainsi d'usage d'utiliser des procédures de recherche pas à pas, sous-optimales :

1. Sélection montante. On initialise le sous-ensemble avec une seule variable (celle qui maximise le critère). On ajoute ensuite une variable, on obtient une nouvelle valeur pour le critère. On procède ainsi pour toutes les variables et on sélectionne la variable qui a fait le plus augmenter le critère (si on cherche à maximiser le critère). On obtient un sous-ensemble de taille 2, puis on recommence l'ajout des variables une à une jusqu'à atteindre la taille de sous-ensemble souhaitée.
2. Sélection descendante. On initialise avec le sous-ensemble complet, on enlève une à une les variables et on calcule la valeur du critère sur le sous-ensemble retenu. On sélectionne le sous-ensemble qui maximise le critère et on recommence jusqu'à atteindre la taille de sous-ensemble souhaitée.
3. Sélection descendante/montante, qui est une combinaison des deux précédentes.

Ces procédures sont plus ou moins gloutonnes selon que le calcul (unitaire) du critère est important et que le nombre de configurations testées est élevé. De plus, cela dépendra de la réutilisation ou non de calculs déjà effectués. Il est aussi nécessaire de définir un critère d'arrêt pour ces procédures. Dans la plupart des cas, le critère d'arrêt repose simplement sur le nombre d'électrodes du sous-ensemble.

Sélection embarquée. Il existe également une autre manière de réaliser une sélection qui permet de choisir un ensemble de variables au cours de l'apprentissage des classifieurs ou des

filtres spatiaux. Ces méthodes sont dites embarquées car la sélection est effectuée par l'algorithme de classification. Ce type de sélection est généralement une conséquence de l'emploi d'une méthode parcimonieuse (comme pour la régression logistique, cf. section 3.3.4).

Limitations. Les méthodes de sélection d'électrodes de l'état de l'art se reposent généralement sur des critères en bout de chaîne comme le résultat d'une classification en validation croisée [61, 109] rendant ainsi la procédure longue et la réutilisation d'une partie des calculs impossible. De plus les critères haut niveau débouchent sur des ensembles sélectionnés entièrement spécifiques à la chaîne de traitement utilisée et donc largement non optimaux si l'on change un paramètre de la chaîne. Il est alors dans notre intérêt de définir un critère efficace et bas niveau pour obtenir une sélection qui pourrait être effectuée en amont et qui resterait valide quelle que soit la chaîne de traitement utilisée.

Enfin, dans l'optique d'une utilisation entièrement automatisée, c'est-à-dire sans avoir de paramètres à régler, il serait intéressant de disposer d'une méthode de sélection qui ajuste d'elle-même ses paramètres, principalement le nombre d'électrodes à sélectionner.

6.2 Sélection basée sur un critère de détermination

Nous avons ainsi proposé une méthode originale de sélection d'électrodes basée sur le calcul du coefficient de détermination entre les variances des différentes électrodes et la classe des tests. Ce travail a fait l'objet d'une communication lors de la conférence GRETSI 2009 [5].

6.2.1 Méthode

Notre sélection de modèle repose sur l'emploi du coefficient de détermination R^2 qui reflète la force du lien entre les log-variances des électrodes et la classe du test. La variable à expliquer est la classe y_i du test et les variables explicatives sont par convention les log-variances de chacune des C électrodes. On utilise le vecteur constitué de la log-variance des électrodes pour calculer le coefficient de détermination. Ainsi, notons $v_{i,c}$ la composante i du vecteur \mathbf{v}_c , égale à la log-variance calculée sur le signal de l'électrode $c \in \{1 : C\}$ pendant le test $i \in \{1 : I\}$. On a :

$$\mathbf{v}_c = [v_{1,c} \dots v_{I,c}]^T \in \mathbb{R}^{I \times 1}$$

Le calcul du coefficient de détermination est le suivant : Soit $\mathbf{A} = [\mathbf{v}_1 \dots \mathbf{v}_C \mathbf{y}] \in \mathbb{R}^{I \times (C+1)}$ la matrice composée de la concaténation de tous les vecteurs de variances et du vecteur \mathbf{y} contenant la classe des différents tests, on calcule $\mathbf{R}_{\mathbf{A}} \in \mathbb{R}^{(C+1) \times (C+1)}$ la matrice de corrélation de \mathbf{A} :

$$[\mathbf{R}_{\mathbf{A}}]_{i,j} = \text{corr}(\mathbf{a}_i, \mathbf{a}_j) \quad (6.1)$$

Chaque éléments de $\mathbf{R}_{\mathbf{A}}$ représente donc la valeur de la corrélation entre 2 vecteurs colonne de \mathbf{A} . La matrice $\mathbf{R}_{\mathbf{A}}$ a une structure particulière que l'on résumera ainsi :

$$\mathbf{R} = \begin{bmatrix} \mathbf{R}_v & \mathbf{r}_{vy} \\ \mathbf{r}_{yv} & 1 \end{bmatrix}$$

avec $\mathbf{R}_v \in \mathbb{R}^{C \times C}$ la matrice de corrélation inter-électrodes et $\mathbf{r}_{vy} = \mathbf{r}_{yv}^T \in \mathbb{R}^C$ le vecteur des corrélations entre les électrodes et la classe des tests. On obtient alors le coefficient de détermination R^2 :

$$R^2 = \mathbf{r}_{vy}^T \mathbf{R}_v^{-1} \mathbf{r}_{vy}$$

Ce coefficient de détermination mesure la part de variance expliquée par le modèle. En d'autres termes, il témoigne de la qualité de la régression entre nos caractéristiques (log-variance sur chaque électrode) et la classe des essais. Ainsi, on cherche à sélectionner un modèle comportant un nombre minimum d'électrodes qui explique le mieux la classe des essais. Pour pouvoir comparer des modèles avec un nombre variable k de paramètres, i.e. d'électrodes, on utilise dans notre étude le coefficient de détermination ajusté [114] qui permet de comparer des modèles dont le nombre de variables n'est pas constant :

$$R_a^2 = 1 - \frac{I - 1}{I - (c + 1)}(1 - R^2)$$

avec I le nombre de tests et $c \leq C$ le nombre d'électrodes considérées. La procédure proposée se déroule en deux étapes.

Dans un premier temps, on recherche pour un nombre donné c d'électrodes, le meilleur sous-ensemble S_c^* de c électrodes parmi les C électrodes qui vérifie :

$$S_c^* = \operatorname{argmax}_{S_c} R_a^2(S_c)$$

Cette optimisation est faite de manière non exhaustive par une technique modifiée de régression pas à pas ascendante (forward).

Dans un second temps, on détermine le nombre minimal d'électrodes C^* qui minimise le critère d'information d'Akaike corrigé [1] sur l'ensemble des sous-ensembles $\{S_c^*\}_{c=\{1:C\}}$:

$$C^* = \operatorname{argmin}_c AIC_{cor}(c)$$

avec

$$AIC_{cor}(c) = I \ln(\hat{\sigma}^2(c)) + 2c + \frac{2c(c+1)}{I - c - 1}$$

$$\text{avec } \hat{\sigma}^2(c) = \frac{1}{I} \sum_{i=1}^I \varepsilon_i^2(c)$$

où $\hat{\sigma}^2(c)$ est une estimation de la variance des termes résiduels $\varepsilon_i(k)$ de notre modèle bâti sur le sous-ensemble d'électrodes S_c^* . La variance des termes résiduels est liée à $R_a^2(S_c^*)$ par :

$$\hat{\sigma}^2(c) = (1 - R_a^2(S_c^*))\sigma_y^2$$

avec σ_y^2 la variance des y_i .

Variante Robuste. Les estimations de variance seront réalisées de manière avantageuse par des techniques empruntées aux statistiques robustes. En effet, les signaux EEG sont souvent contaminés par des artefacts. Ainsi, la variance des électrodes sera estimée par la déviation absolue moyenne de la médiane (MAD) [130].

6.2.2 Illustration

Ce travail sera illustré sur l'utilisateur *al* du jeu de données IVa de la compétition BCI III. On dispose de 118 électrodes, et on cherche à obtenir un sous-ensemble d'électrodes de taille minimale qui conserve le maximum d'information vis-à-vis des 2 tâches mentales réalisées : à savoir un mouvement imaginé de la main et du pied.

On applique donc la procédure décrite précédemment. Suite à la sélection ascendante, on dispose de valeur R_a^2 maximale et de AIC_c minimale pour chaque taille de sous-ensemble. On peut alors observer l'évolution de la valeur de coefficient de détermination ajusté et du

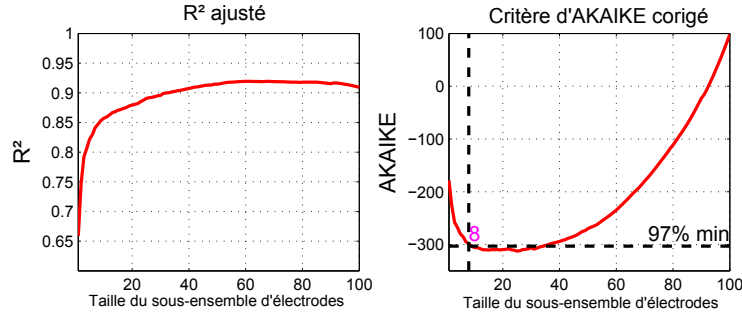


FIGURE 6.1 – Évolution du coefficient de détermination ajusté et du critère d’Akaike corrigé en fonction de la taille du sous-ensemble d’électrodes.

critère d’Akaike corrigé en fonction du nombre d’électrodes considérées comme l’illustre la figure 6.1.

Contrairement au R^2 classique, le coefficient de détermination ajusté prend en compte la dimension du problème, la fonction n’est pas monotone. En effet si la dimension du problème augmente pour atteindre des valeurs élevées, la complexité introduite par la dimension supplémentaire va dépasser le gain en corrélation. L’introduction d’une nouvelle variable fait alors chuter la valeur du R^2 ajusté. On observe un maximum du R^2_a pour une taille de sous-ensemble de 60 électrodes environ. De même, la croissance est très forte pour la première dizaine d’électrodes puis se réduit rapidement. On peut ainsi affirmer que la majeure partie de la corrélation est portée par un nombre relativement faible d’électrodes. Dès lors, on peut facilement imaginer qu’il existe un sous-ensemble d’électrodes quasiment optimal qui offrirait de bonnes performances mais ne possédant pas un R^2_a maximal. Intuitivement, on pourrait placer ce point au niveau de l’inflexion de la courbe du R^2_a c’est-à-dire autour de la dizaine d’électrodes.

Le critère d’Akaike nous fournit une critère objectif dans la recherche de la taille optimale du sous-ensemble d’électrodes. Comme on le voit sur la figure 6.1, le critère d’Akaike forme un plateau en son minimum, plutôt que de sélectionner le minimum absolu (qui peut varier de manière importante), il est préférable de retenir le premier point qui atteint un certain pourcentage du minimum (ici 97%).

La taille optimale retenue dans ce cas est de $k^* = 8$ électrodes. La figure 6.2 donne la position de ce sous-ensemble d’électrodes ainsi que la principale caractéristique spatiale. Les électrodes sélectionnées se situent globalement au niveau des zones motrices, et la caractéristique spatiale conserve la même topologie.

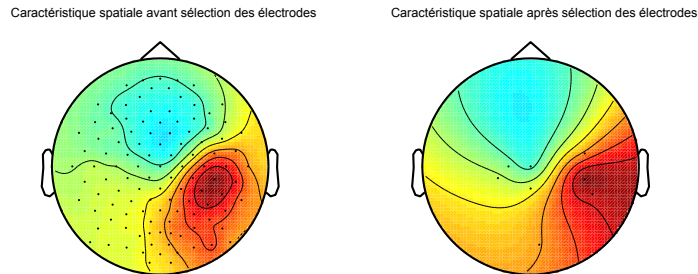


FIGURE 6.2 – Position des électrodes et caractéristiques spatiales avant et après sélection d’électrodes.

Enfin, on peut vérifier à la figure 6.3 que le sous-ensemble sélectionné est bien de taille optimale en observant les performances finales en validation croisée (taux de classification correcte). Notre méthode montre également de meilleures performances qu'un algorithme similaire de l'état de l'art basé sur un critère d'information mutuelle [68].

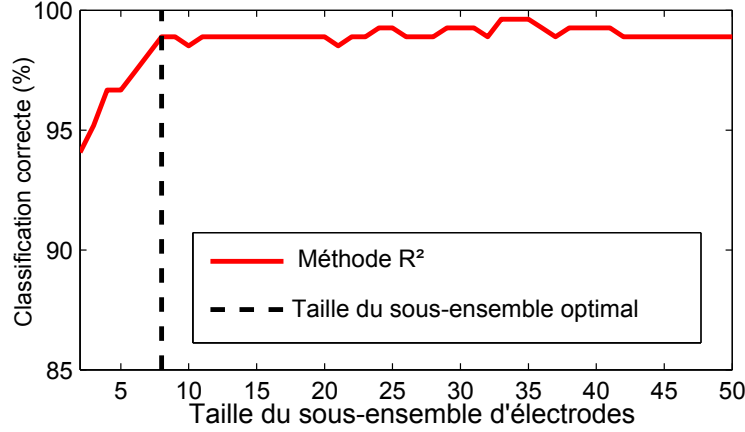


FIGURE 6.3 – Évolution du taux de classification correcte en fonction de la taille du sous-ensemble

On remarque clairement que le sous-ensemble sélectionné est celui de taille minimum conservant une performance en classification pratiquement égale à la performance maximale.

6.3 Sélection basée sur un critère de géométrie Riemannienne

L'idée générale de cette procédure de sélection d'électrodes est d'utiliser le critère de distance de Riemann entre les centres des classes afin d'évaluer les électrodes porteuses d'information (i.e. utiles à la séparabilité de chaque classe). On peut considérer qu'une électrode ne porte pas d'information à partir du moment où l'introduction de l'électrode en question n'augmente pas la distance entre les centres des classes.

La figure 6.4 illustre ce concept. On voit que sur l'électrode n°1 la distance entre les classes est plus importante que sur l'électrode n°2. L'électrode n°2 est donc "moins importante" que l'électrode n°1 pour le discrimination des deux classes.

6.3.1 Méthode

La méthode sera présentée pour un cas à deux classes. On définit donc le critère qui va orienter la recherche de notre sélection d'électrode grâce à la distance Riemannienne entre les matrices de covariance moyennes. Soit $\Sigma^{(1)}$ et $\Sigma^{(2)}$ les 2 matrices de covariance moyennes calculées à partir de l'ensemble des matrices de covariance de chaque classe, on a :

$$J = \delta_R(\Sigma^{(1)}, \Sigma^{(2)}) \quad (6.2)$$

Ce critère se généralise aisément à des cas multi-classes en calculant la somme des distances entre les paires des centres de chaque classe :

$$J = \sum_{k=1}^K \sum_{j>k}^K \delta_R(\Sigma^{(k)}, \Sigma^{(j)}) \quad (6.3)$$

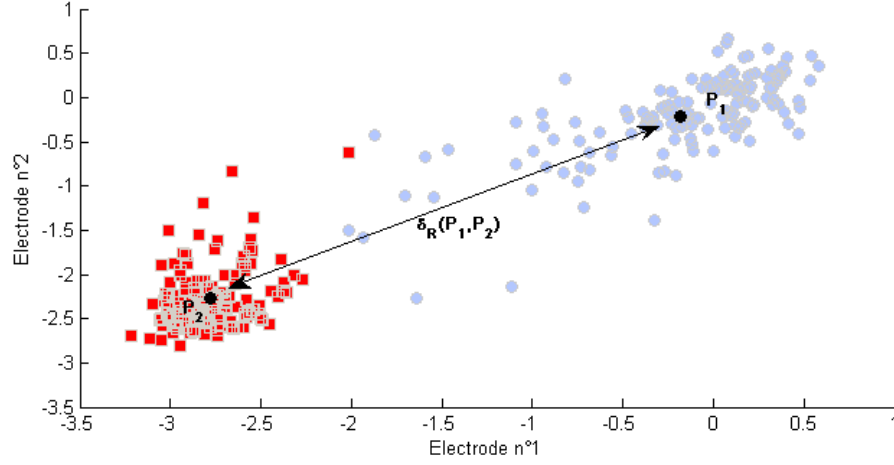


FIGURE 6.4 – Illustration schématique (cas non réel) de la distance entre les centres des classes

On applique ensuite une procédure descendante pour construire notre sous-ensemble d'électrodes. Pour les matrices de covariance, chaque ligne/colonne d'indice i représente l'information portée par l'électrode i . En effet, pour chaque élément de la matrice de covariance $[\Sigma]_{i,j} = \text{cov}(\mathbf{x}_i, \mathbf{x}_j)$ avec \mathbf{x}_i et \mathbf{x}_j les signaux temporels issus des électrodes i et j .

En supprimant la i -ème ligne et la i -ème colonne des matrices de covariance intra-classe, on retire du sous-ensemble l'électrode $n^\circ i$. Cette technique a l'avantage de ne pas nécessiter de recalculer les matrices de covariance intra-classe à chaque fois qu'on modifie le sous-ensemble ce qui constitue un gain non négligeable en temps de calcul. En effet, quand les méthodes classiques utilisent des caractéristiques de haut niveau (par exemple après filtrage spatial), chaque modification du sous-ensemble d'électrodes impactant la suite de la chaîne, il est nécessaire de recalculer l'ensemble des paramètres (ex : recalculer les filtres spatiaux). On définit ainsi l'opérateur $\text{reduce}(\Sigma, i)$ qui retire de la matrice de covariance les éléments correspondants à l'électrode d'indice i .

La procédure descendante est résumée par l'algorithme 4.

Supposons qu'un sous-ensemble de N électrodes ait été sélectionné. Pour connaître l'électrode à retirer de ce sous-ensemble, on retire séquentiellement une des électrodes de chacune des matrices de covariance moyennes :

$$\begin{aligned}\tilde{\Sigma}^{(1)} &= \text{reduce}(\Sigma^{(1)}, i) \\ \tilde{\Sigma}^{(2)} &= \text{reduce}(\Sigma^{(2)}, i)\end{aligned}\tag{6.4}$$

On calcule ensuite le critère à partir de ces matrices réduites :

$$J(i) = \delta_R(\tilde{\Sigma}^{(1)}, \tilde{\Sigma}^{(2)})\tag{6.5}$$

On effectue indépendamment cette opération pour chacune des N électrodes du sous-ensemble, on obtient donc N valeurs pour le critère. L'électrode que l'on doit retirer du sous-ensemble est l'électrode qui maximise le critère. En effet, l'électrode dont l'absence maximise le critère est celle qui contribue le moins à la distance totale entre les deux classes.

$$i^* = \underset{i}{\operatorname{argmax}} J(i)\tag{6.6}$$

Une fois l'électrode d'indice i^* retirée, on dispose d'un sous-ensemble de taille $N - 1$, et on recommence la procédure jusqu'à obtenir un sous-ensemble de taille C^* .

Algorithm 4 Algorithme de sélection d'électrodes basée sur la distance riemannienne

Entrée : $\Sigma^{(1)}$ et $\Sigma^{(2)}$ les matrices de covariance moyenne des deux classes

Entrée : C^* le nombre d'électrodes que l'on souhaite sélectionner

Sortie : *Subset* la liste de indices des électrodes sélectionnées

```

1: Subset = [1 ... C] {Initialisation du sous-ensemble}
2: for  $c = 1$  to  $C - C^*$  do
3:   for  $i = 1$  to  $C - c + 1$  do
4:      $\tilde{\Sigma}^{(1)} = \text{reduce}(\Sigma^{(1)}, i)$  {On retire temporairement l'électrode  $i$ }
5:      $\tilde{\Sigma}^{(2)} = \text{reduce}(\Sigma^{(2)}, i)$ 
6:      $J(i) = \delta_R(\tilde{\Sigma}^{(1)}, \tilde{\Sigma}^{(2)})$  {Calcul du critère sur les matrices réduites}
7:   end for
8:    $i^* = \text{argmax}_i J(i)$  {sélection de l'indice qui maximise le critère}
9:    $\Sigma^{(1)} = \text{reduce}(\Sigma^{(1)}, i^*)$  {On retire définitivement l'électrode}
10:   $\Sigma^{(2)} = \text{reduce}(\Sigma^{(2)}, i^*)$ 
11:  Subset( $i^*$ ) = [ ] {On retire l'électrode du sous-ensemble}
12: end for
13: return Subset

```

Cette méthode de sélection a le grand avantage d'être très bas niveau, c'est-à-dire qu'elle peut s'appliquer à des données brutes. De plus elle n'est pas dépendante d'une méthode particulière comme cela est le cas pour les méthodes de sélection reposant sur les coefficients des filtres spatiaux [125, 78]. Elle reste donc valide pour tout type de chaîne de traitement employée ultérieurement. Utilisée en combinaison avec le filtrage spatial par CSP, elle est extrêmement efficace. En effet le critère qu'elle cherche à optimiser est la distance entre les centres des classes, critère qui est également indirectement utilisé dans le calcul des filtres spatiaux par CSP. Cette méthode a fait l'objet d'un dépôt de brevet et également d'une communication à la conférence *Neural Engineering 2011* [6].

Sélection spécifique à l'application. Afin de sélectionner un sous-ensemble spécifique à l'application, on utilisera les sous-ensembles obtenus sur chacun des sujets. Pour un nombre C^* d'électrodes dans les sous-ensembles, on compte le nombre de fois où chaque électrode est sélectionnée. On trie alors les électrodes et on conserve les C^* qui reviennent le plus souvent dans les différents sous-ensembles spécifiques à chaque utilisateur. S'il y a une indétermination (différentes électrodes qui reviennent un même nombre de fois), on utilise alors les sous-ensembles de taille $C^* + 1$ en y appliquant la même procédure, jusqu'à ce que l'indétermination soit levée. Cette procédure n'est pas spécifique à cette méthode.

Variante. L'emploi de la distance au centre seule est un critère sous-optimal dans le sens où il ne prend pas en compte la dispersion des classes. On peut considérer qu'il reste valide en tant qu'approximation de la séparabilité des classes, si la dispersion reste globalement égale pour toutes les électrodes. Pour être plus rigoureux il faut prendre en compte cette dispersion. L'utilisation d'un t-test semble appropriée à la problématique. Le t-test classique est défini comme ce qui suit :

$$t = \frac{\mu^{(1)} - \mu^{(2)}}{\sqrt{\frac{\sigma^2(1)}{I^{(1)}} + \frac{\sigma^2(2)}{I^{(2)}}}} \quad (6.7)$$

avec $\mu^{(k)}$, $\sigma^{(k)}$ et $I^{(k)}$ respectivement la moyenne, l'écart type et le nombre de tests de la classe k . Il faut ainsi définir la variance dans l'espace de Riemann. Par extension directe

de la variance dans le cas euclidien, on définit la variance dans la variété par l'espérance des distances au carrée entre chaque test et la moyenne.

$$(s_R^{(k)})^2 = \frac{1}{I^{(k)}} \sum_{i=1}^{I^{(k)}} \delta_R(\Sigma_i, \Sigma^{(k)})^2 \quad (6.8)$$

Finalement, le t-test défini dans Riemann est donné par :

$$t_R = \frac{\delta_R(\Sigma^{(1)}, \Sigma^{(2)})}{\sqrt{\frac{(s_R^{(1)})^2}{I^{(1)}} + \frac{(s_R^{(2)})^2}{I^{(2)}}}} \quad (6.9)$$

Cette variante augmente grandement le coût calculatoire de la méthode, mais elle est indispensable lorsque la dispersion des données varie grandement entre les différentes électrodes. C'est par exemple le cas lorsqu'on sélectionne des électrodes avec des filtrages fréquentiels différents.

6.3.2 Illustration et résultats

Le jeu de données utilisé est le IVa de la compétition BCI III. Il s'agit d'un jeu de données à deux classes (mouvement imaginé de la main droite et du pied) enregistré sur 118 électrodes.

Nous utiliserons la version **sans t-test** de cette méthode. La figure 6.5 montre l'évolution du critère (6.2) en fonction de la taille du sous-ensemble d'électrodes sélectionné. Le critère est normalisé par sa valeur maximale, c'est-à-dire en le divisant par la distance entre les deux matrices de covariance moyennes calculées avec l'ensemble des 118 électrodes.

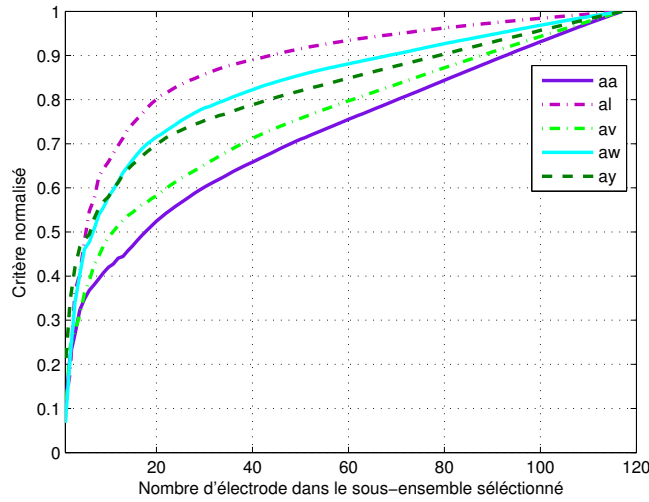


FIGURE 6.5 – Évolution du critère Eq.(6.2) normalisé, en fonction de la taille du sous-ensemble d'électrodes sélectionnées.

Pour chacun des 5 sujets, on observe le même comportement : une évolution rapide du critère suivie d'une inflexion pour finir sur une croissance linéaire. Cette dernière nous informe que les dernières électrodes ajoutées sont équivalentes en terme de distance et ne sont donc pas pertinentes pour la discrimination des deux tâches mentales. On observe également qu'un faible nombre (< 20) d'électrodes portent la majeure partie de la distance

entre les deux classes. Pour le sujet *al*, le sous-ensemble de 20 électrodes supporte 80% de la distance totale.

La figure 6.6 présente le résultat de la sélection de 10 électrodes pour les 5 sujets, ainsi que la sélection spécifique à l'application correspondante. Les 10 électrodes sont globalement positionnées au-dessus du cortex sensori-moteur, ce qui est en accord avec les tâches mentales impliquées. En particulier la sélection spécifique à l'application le montre encore plus clairement, avec une majorité d'électrodes au dessus des zones dédiées aux mouvements des mains.

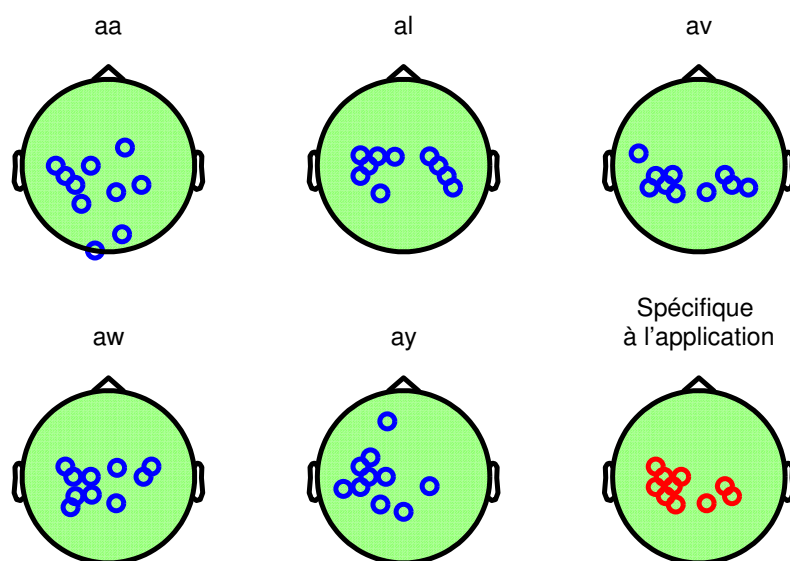


FIGURE 6.6 – Sélection de 10 électrodes pour les 5 sujets, ainsi que la sélection spécifique à l'application.

La figure 6.7 montre l'évolution des performances de la classification en fonction du nombre d'électrodes. La chaîne de traitement employée est la chaîne standard décrite dans le chapitre 3 et constituée d'un filtrage spatial par CSP et d'une classification par LDA. 3 paires de filtres sont utilisées, comme il est d'usage avec ce type de traitement.

Comme on peut l'observer, la procédure de sélection atteint avec succès une performance maximale avec un nombre d'électrodes très réduit. A titre d'exemple, on atteint 97% de bonne classification avec seulement 3 électrodes pour le sujet *al* (il en fallait 8 pour atteindre la même performance avec la méthode basée sur la corrélation présentée dans la section 6.2). En moyenne (courbe avec les points noirs) un sous-ensemble de 10 électrodes semble suffisant pour atteindre les performances optimales. De plus, on pourra remarquer que la valeur du critère (6.2) proposé pour la sélection est fortement corrélé à la performance en classification étant donné que les courbes présentées aux figures 6.5 et 6.7 sont pratiquement triées dans le même ordre. Ce résultat n'est pas étonnant compte tenu du fait que le critère utilisé ici est fortement lié au critère optimisé par la CSP.

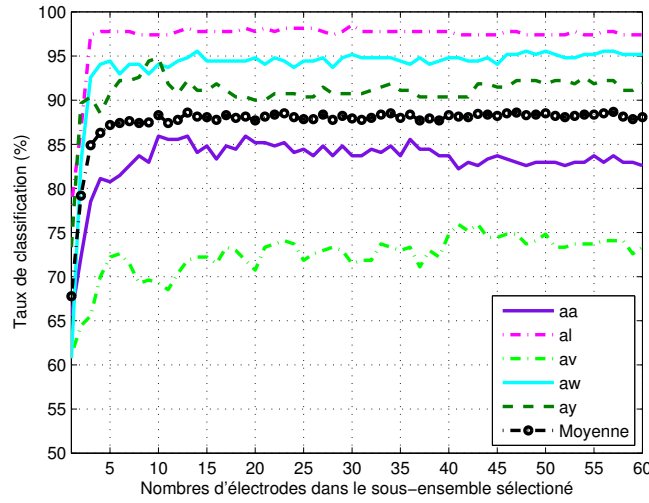


FIGURE 6.7 – Résultat de classification en fonction du nombre d'électrodes sélectionnées, 30-fold cross-validation.

6.4 Résultats comparatifs

Introduction. Cette section présente les résultats numériques pour les différentes méthodes comparées :

1. **CSP** : l'ensemble des électrodes sont utilisées.
2. **Coeff** : la sélection des électrodes est basée sur la valeur des coefficients des principaux filtres spatiaux. Le principe étant qu'une électrode est d'autant plus importante que la valeur (absolue) de son coefficient est forte. Cette méthode est décrite dans [125].
3. **MI** : la sélection est réalisée suivant une procédure ascendante. Cette procédure est similaire à notre procédure de sélection fondée sur le critère de corrélation multiple, mais avec l'emploi d'un critère fondé sur l'information mutuelle entre la variance des signaux et la classe des test. C'est une implémentation de l'algorithme décrit dans [68].
4. R^2 : sélection basée sur un critère de corrélation, procédure ascendante. C'est la première méthode proposée dans ce chapitre.
5. **Riemann** : Sélection basée sur un critère de distance Riemannienne entre les centres des classes couplée à une procédure descendante. C'est la seconde méthode proposée dans ce chapitre. La version sans t-test est utilisée.

L'algorithme de sélection est appliqué après un filtrage passe-bande (Butterworth d'ordre 5) entre 8 et 35 Hz. La chaîne de traitement qui s'en suit est constituée d'un filtrage spatial par CSP et d'une classification par LDA de Fisher. On conserve uniquement les 3 meilleures paires de filtres spatiaux comme il est d'usage.

Performances dans les conditions de la compétition : Dans un premier temps, nous allons nous placer dans les conditions de la compétition BCI III. Ce jeu de données propose un nombre de réalisations dans la base d'apprentissage différent pour chaque utilisateur. Ainsi, sur les 280 réalisations disponibles, on utilisera comme base d'apprentissage les 168, 224, 84, 56 et 28 premières pour respectivement les utilisateurs *aa*, *al*, *av*, *aw* et *ay*. La taille de la base d'apprentissage est cruciale pour la CSP. En particulier quand le nombre d'électrodes est élevé, on obtient généralement une mauvaise estimation des filtres spatiaux. Diminuer efficacement le nombre d'électrodes permet ainsi de palier à ce genre de problème.

Les résultats sont reportés Table 6.1, et sont évalués pour les 4 méthodes de sélection pour un nombre fixe de 10 électrodes sélectionnées.

Sujet	CSP	Coeff	MI	R ²	Riemann
<i>aa</i>	69.6	75.9	69.6	75	81.3
<i>al</i>	96.4	98.2	94.6	100	98.2
<i>av</i>	52	49	61.2	61.7	65.3
<i>aw</i>	79	75.9	61.2	68.3	78.1
<i>ay</i>	50.4	77.4	84.5	84.5	86.5
Moyenne	69.5	75.3	74.2	77.9	81.9

TABLE 6.1 – Performances en pourcentage de classification correcte pour les différentes méthodes de sélection. 10 électrodes sont sélectionnées avec une sélection spécifique à chaque utilisateur.

L'utilisation de l'ensemble des électrodes conduit à une performance moyenne de 70% environ. Pour deux sujets (*av* et *ay*), la classification est inefficace (50% d'erreur). Ceci est dû au très faible nombre de données dans la base d'apprentissage (28 pour *ay*) qui rend les filtres spatiaux et la classification inopérants.

Les deux méthodes de l'état de l'art (Coeff et MI) offrent des résultats similaires de l'ordre de 75% en moyenne. L'amélioration est plus particulièrement effective pour les deux sujets (*av* et *ay*) où la réduction du nombre d'électrodes diminue l'impact de la taille de la base d'apprentissage. La méthode fondée sur la corrélation (R²) atteint 78% de classification correcte en moyenne.

Enfin, la méthode reposant sur la géométrie Riemannienne montre les meilleurs résultats, avec un score moyen de 82%. L'amélioration est claire pour tous les sujets, sauf le sujet *aw* dont les performances sont similaires à la méthode de référence (CSP). En comparaison des résultats obtenus par les participants de la compétition, ce résultat se classe juste derrière la 3ème place et ce uniquement en réduisant le nombre d'électrodes.

Un des avantages de la méthode R² est qu'elle propose un moyen de sélectionner automatiquement le nombre d'électrodes. Ainsi la table 6.2 montre le résultat de cette sélection automatique en termes de performance et de nombre d'électrodes sélectionnées. On observe des performances similaires (bien que légèrement meilleures) par rapport au cas où le sous-ensemble était de taille fixe (10 électrodes). Le nombre d'électrodes retenues varie entre 4 pour le sujet *aa* et 9 pour *ay*, avec une moyenne de 6 électrodes environ. La sélection automatique est donc réellement efficace et conduit aux performances optimales pour la méthode en question.

User	# Elec	Performance (%)
<i>aa</i>	4	75
<i>al</i>	6	92.9
<i>av</i>	5	63.8
<i>av</i>	5	74.1
<i>ay</i>	9	87.3
Moyenne	5.8	78.6

TABLE 6.2 – Nombre d'électrodes et performance de la sélection R² quand la taille du sous-ensemble d'électrodes est sélectionnée automatiquement. Seuil placé à 97% du minimum du critère d'Akaike.

Enfin, on pourra également observer les performances de la sélection spécifique à l'application pour les différentes méthodes, données table 6.3. Pour chaque sujet, un sous-ensemble d'électrodes est construit à partir de sous-ensembles obtenus sur les 4 autres sujets.

User	Coeff	MI	R^2	Riemann
<i>aa</i>	68.8	62.5	70.5	74.1
<i>al</i>	87.5	85.7	89.3	94.6
<i>av</i>	73.5	55.6	60.2	66.8
<i>aw</i>	79.0	62.5	77.7	68.3
<i>ay</i>	71.4	73.4	73.8	81.3
moyenne	76	67.9	74.3	77

TABLE 6.3 – Performances en classification pour les différentes méthodes de sélection dans le cadre de la sélection spécifique à l'application. 10 électrodes sélectionnées

Il est intéressant de remarquer que la méthode qui s'en sort le mieux est celle fondée sur les coefficients des filtres spatiaux, avec une très légère augmentation des performances. En particulier pour le sujet *av* qui passe ainsi de 49% à 73.5% de bonne classification. Pour l'ensemble des autres sujets, on observe une diminution des performances par rapport à la sélection spécifique au sujet. L'utilité de la sélection spécifique à l'application est donc évidente dans certains cas où les données d'apprentissage semblent être corrompues. En revanche, on ne dispose ici que de trop peu de sujets pour obtenir une sélection réellement généralisable.

La méthode de sélection par géométrie riemannienne améliore légèrement les résultats sur le sujet *av* mais perd globalement sur l'ensemble des sujets. Elle offre néanmoins les meilleurs résultats en comparaison des autres méthodes.

Performances en validation croisée. Une procédure de validation croisée (30-fold) est maintenant appliquée pour les différentes méthodes afin de les comparer sans introduire le biais dû à la taille variable de la base d'apprentissage. Ainsi la figure 6.8 montre les performances en classification en fonction de la taille du sous-ensemble sélectionné.

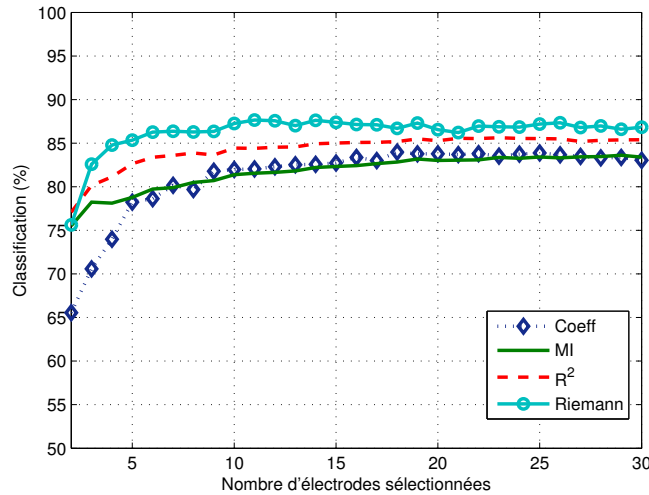


FIGURE 6.8 – Résultat de classification en fonction du nombre d'électrodes sélectionnées pour les différentes méthodes présentées, 30-fold cross-validation.

Pour un sous-ensemble de deux électrodes, les méthodes MI, R^2 et Riemann ont les mêmes performances, ce qui tend à montrer que les sous-ensembles sélectionnés sont équivalents. Dès que la taille du sous-ensemble commence à augmenter, la méthode Riemann se démarque

très nettement des autres. Les 4 méthodes finissent par converger en performance autour de 85% pour des tailles de sous-ensembles élevées.

Le classement des 4 méthodes est donc le suivant. La méthode Coeff arrive en dernière position, bien qu'elle soit rapidement équivalente à la méthode MI, elle éprouve plus de difficultés pour les sous-ensembles inférieurs à 5 électrodes. Vient ensuite la méthode MI, surpassée par la méthode R^2 qui offre dans tous les cas des performances plus élevées. Enfin, la méthode Riemann montre les meilleures performances, dépassant les 85% dès la 4^{ième} électrode.

6.5 Conclusion

Dans la réalisation d'une ICM, réduire le nombre d'électrodes présente de nombreux avantages (accélérer la mise en place du système EEG, diminuer la gêne de l'utilisateur, réduire le coût calculatoire des algorithmes et diminuer le coût de l'équipement). En dépit de ces avantages, cette procédure est généralement peu utilisée car les IMC sont principalement réalisées dans un cadre de recherche où l'on cherche avant tout à acquérir des signaux EEG sur un maximum d'électrodes pour réaliser des traitements ultérieurs. En revanche, dans une application commerciale, où l'ergonomie, le coût et la robustesse sont des éléments particulièrement importants, l'utilisation d'une procédure de sélection prend tout son sens.

Ce chapitre a présenté deux méthodes originales de sélection. L'une, basée sur un critère de corrélation multiple entre les signaux EEG et la classe des réalisations, possède l'avantage de sélectionner automatiquement et de manière optimale le nombre d'électrodes à retenir dans le sous-ensemble. L'autre, basée sur un critère de distance riemannienne entre les centre des classes, est très performante et permet de sélectionner un sous-ensemble de taille minimale sans perte d'information. En outre, cette dernière méthode s'applique directement après le filtrage fréquentiel, ce qui ne contraint pas le sous-ensemble sélectionné à être utilisé avec un type de traitement particulier.

La sélection d'électrode spécifique à chaque sujet permet d'obtenir de meilleurs résultats en moyenne. En revanche la sélection spécifique à l'application offre une bonne alternative pour les cas difficiles. Cette dernière est également un bon compromis en terme de performance si on ne souhaite pas réaliser une première session avec beaucoup d'électrodes pour chaque sujet, nécessaire à l'application d'une sélection spécifique à l'utilisateur.

Enfin, on pourra également appliquer ces algorithmes pour sélectionner des couples bande de fréquences / électrode simplement en découpant les signaux originaux par un banc de filtres. En revanche, on augmentera considérablement le nombre d'électrodes (artificielle) ce qui pourra poser des problèmes de sur-apprentissage.

Troisième partie

Contributions Expérimentales

Mise en place expérimentale d'une ICM

Ce chapitre porte sur la mise en place des expérimentations réalisées au sein du laboratoire LE2S. Il s'agit de la réalisation d'un brain-switch c'est-à-dire un interrupteur commandé par activation cérébrale volontaire. Cette réalisation a été guidée à la fois par la robustesse de l'ICM et par l'optimisation de l'apprentissage du sujet.

7.1 Chaîne expérimentale EEG

La mise en place d'une chaîne expérimentale pour les ICMs est une tâche plus complexe qu'il n'y paraît. Avant toute chose, il faut se questionner sur le but que l'on souhaite atteindre. On distingue principalement deux motivations :

Pour le développement algorithmique. Une première motivation est de constituer une base de données dont on maîtrise le processus d'acquisition et qui sera ensuite utilisée pour la mise au point d'algorithmes. On va alors se placer dans les conditions les plus favorables, avec notamment une salle d'expérimentation optimisée pour réduire le bruit électromagnétique (pas ou peu de matériel électronique, pas d'éclairage néons etc ..). Ce type d'expérimentation est couramment réalisée dans le domaine de la neuro-physiologie pour l'étude des processus cognitifs. Si ce type d'expérimentation est indispensable, elle ne reflète que très peu une situation réelle d'utilisation d'une ICM.

Pour une mise en situation réelle : Une seconde motivation est de tester un système pouvant fonctionner en condition de vie réelle, que cela soit pour un démonstrateur ou en déploiement auprès de sujets (handicapés ou non). Dans ce cas, on fait apparaître des contraintes fortes et des problématiques nouvelles. Ainsi la qualité des données n'est plus un paramètre maîtrisé et devient une contrainte. En effet, on ne connaît pas l'environnement dans lequel les signaux seront enregistrés et on ne peut pas intervenir dessus, on ne peut pas contraindre l'utilisateur à ne pas bouger ou à ne pas laisser sa pensée divaguer. Il n'est pas non plus question de passer un temps important à la mise en place des électrodes, ce qui diminue automatiquement le nombre maximal de ces dernières ainsi que la qualité de la pose. De plus, il faut prendre en compte la fatigue de l'utilisateur et parfois la possibilité qu'il se trompe ou ne réalise pas la bonne tâche mentale au bon moment. Il devient de plus en plus difficile de disposer d'une vérité de terrain. Finalement, ces expérimentations ont pour but d'étudier et d'améliorer l'**ergonomie** et la **robustesse** des ICMs et des traitements associés.

Cette thèse s'inscrit entièrement dans cette deuxième approche, avec comme but de tenter de faire sortir les ICMs des laboratoires. Cependant, le processus de réalisation d'expérimentations en situation réelle nécessite de maîtriser les bases de la mesure EEG, les deux types d'expérimentations ont donc été réalisées au laboratoire.

7.1.1 Matériel

La chaîne expérimentale minimale est composée de 3 éléments : l'amplificateur EEG, les électrodes et un ordinateur doté de la suite logicielle nécessaire au traitement des données.

L'amplificateur EEG. L'amplificateur EEG utilisé, dont on peut voir une photo figure 7.1, est l'amplificateur g.USBamp de la société autrichienne g.tec¹. Cet amplificateur possède 16 voies d'enregistrement séparées en 4 groupes de 4 voies, chaque groupe possède des entrées dédiées pour la masse et pour la référence, ces dernières pouvant également être communes à l'ensemble des voies. Chaque voie est amplifiée puis numérisée sur 24 bits. La plage de mesure est de ± 250 mV ce qui résulte d'une résolution en tension inférieure au μ V. La fréquence d'échantillonnage en sortie de l'amplificateur peut être réglée entre 64 Hz et 38.4 kHz mais l'amplificateur effectue un sur-échantillonnage à la fréquence de 2.45 MHz ce qui permet d'obtenir un excellent rapport signal sur bruit. De plus, l'amplificateur intègre un DSP² lui permettant de réaliser un filtrage numérique du signal avant envoi des données à l'ordinateur. Ce filtrage est réglable et permet de sélectionner une bande de fréquence et/ou de couper la composante à 50 Hz issue de la tension secteur par laquelle les biosignaux sont en permanence perturbés. Enfin, il permet la mesure de l'impédance des électrodes.

Cet amplificateur est de très bonne qualité et est largement sur-dimensionné pour les problèmes rencontrés en ICMs, dont la fréquence d'échantillonnage dépasse rarement le kHz. Dans nos expérimentations, nous utilisons généralement une fréquence d'échantillonnage de 512Hz.



FIGURE 7.1 – L'amplificateur g.USBamp de g.tec.

Les électrodes. Nous disposons de deux jeux d'électrodes, un jeu d'électrodes passives et un jeu d'électrodes actives. Les premières expérimentations se sont déroulées avec le jeu d'électrodes passives. Nous avons rencontré un certain nombre de problèmes avec ces électrodes. En premier lieu, des problèmes d'impédances trop élevées (> 20 k Ω) qui conduisaient invariablement à des signaux de mauvaise qualité et à un grand nombre d'artefacts, générés au moindre mouvement de l'utilisateur (cf section 2.2.4). Nous avons alors concentré nos efforts sur la diminution de cette impédance en soignant la pose des électrodes, notamment à travers une meilleure préparation de la peau du sujet. On utilise généralement une pâte abrasive pour nettoyer la peau au niveau de la surface de contact avec l'électrode.

1. www.gtec.at

2. Digital Signal Processor : un composant de type micro-contrôleur orienté traitement du signal. En général, un DSP possède un grand nombre d'unités de calcul en parallèle.

Cette pâte dégraisse le cuir chevelu et permet d'attaquer la première couche de la peau en retirant les cellules mortes. Cette opération est fastidieuse et peu agréable pour le sujet, il est souvent nécessaire d'insister en particulier si l'impédance est nativement élevée. On doit ensuite écarter les cheveux et appliquer une pâte conductrice entre l'électrode et le cuir chevelu. Pour maintenir le contact, l'électrode est plaquée contre le scalp à l'aide d'un bonnet élastique. Avec ce système, l'utilisateur se sent rapidement oppressé et cela nuit à sa concentration lors de l'expérience.

Cette étape de pose des électrodes, coûteuse en temps et agressive pour le sujet, ne convenait pas aux objectifs d'ergonomie que nous nous sommes fixés. Nous avons alors opté pour des électrodes actives (cf section 2.2.3), qui de par leur nature, fonctionnent de manière efficace à impédance élevée. Ce système est beaucoup plus agréable à porter, puisque d'une part il n'est plus nécessaire de préparer la peau, et d'autre part les électrodes ne sont pas directement en contact avec le cuir chevelu, le contact étant assuré à travers les cheveux par un pont de gel conducteur. Ce pont de gel ayant une certaine plasticité, les électrodes n'ont plus besoin d'être plaquées au scalp de manière stricte. Ainsi le casque qui les maintient est beaucoup moins oppressant pour l'utilisateur. La pose des électrodes est ainsi passée d'environ 30 à 45 minutes pour les électrodes classiques, à moins d'un quart d'heure pour les électrodes actives.

On retrouve des photos de ces deux types d'électrode en figure 7.2.



FIGURE 7.2 – Électrodes passives (à gauche) et actives (à droite).

Suite logicielle : La société g.tec fournit avec son amplificateur un certain nombre de composants logiciels. Nous utilisons celui qui permet d'importer en temps réel les signaux EEG dans l'environnement Simulink de Matlab. Ainsi, nos algorithmes développés sous Matlab peuvent être directement utilisés lors des expérimentations ICM, ce qui accélère leur mise en œuvre. En contrepartie, les interfaces avec l'utilisateur doivent être développées dans le même environnement. Le résultat est fonctionnel mais correspond plus à un processus de prototypage permettant de corriger très rapidement (voire en cours d'expérimentation) la chaîne de traitement. De plus, on maîtrise entièrement la chaîne de l'acquisition à l'affichage. En revanche, il n'est pas possible d'atteindre un degré de finition égal à celui que l'on pourrait obtenir avec l'utilisation d'un logiciel tel qu'OpenViBE. En d'autres termes, la solution logicielle adoptée est parfaite dans le cadre du prototypage, mais montre ses limites dans le cadre d'une utilisation routinière.

En figure 7.3 se trouve une photo de la chaîne expérimentale complète dans une version intermédiaire (électrodes passives).

7.1.2 Remarques générales

La mise en place d'une expérience EEG est délicate, surtout en ce qui concerne la pose des électrodes. Il est absolument nécessaire d'acquérir de l'expérience dans ce domaine en manipulant le plus souvent possible. Les conseils d'expérimentateurs aguerris sont toujours



FIGURE 7.3 – Chaîne expérimentale au laboratoire : l'amplificateur g.USBamp, le casque à électrodes **passives** et l'environnement logiciel avec un écran destiné à l'utilisateur et un écran de contrôle

d'une aide précieuse, nous avons notamment interagi avec le service neurologie du CHU de Grenoble et le Gipsa-Lab.

De plus, il ne faut pas sous-estimer l'impact du matériel sur le sujet. Il est absolument nécessaire de veiller à son confort car ce paramètre aura un impact direct sur sa concentration et donc sur la qualité des données enregistrées.

7.2 Choix du type d'ICM : le *brain-switch*

Cette section présente les objectifs et les choix qui ont guidé la réalisation de notre ICM.

Les objectifs. En plus du développement de méthodes de traitement du signal, le but de cette thèse est de réaliser une interface cerveau machine **asynchrone** et **robuste** à partir de l'EEG. Suite aux premières expériences réalisées sur des paradigmes synchrones (*cue-based*), il est également apparu que l'**ergonomie** est très importante, c'est-à-dire qu'il est nécessaire de prendre en compte le confort du sujet lors de la spécification du paradigme. Nous souhaitons ainsi mettre à contribution de manière plus significative l'utilisateur, lui laissant plus de liberté sur la tâche mentale mais aussi en stimulant le processus d'apprentissage plus intensément, avec en objectif secondaire la réduction du temps de calibrage de l'ICM.

Le cahier des charges fait donc ressortir les propriétés suivantes :

- **Asynchrone** : l'ICM réalisée doit être asynchrone, c'est-à-dire analyser les signaux et fournir une sortie en continu, l'utilisateur pouvant interagir avec elle à tout moment.
- **Robustesse** : l'ICM doit fonctionner dans des conditions réelles (i.e. hors laboratoire) et être la moins perturbée par la présence d'artéfacts ou de points aberrants.

- **Ergonomie** : l'ICM doit être simple à utiliser et la phase d'apprentissage doit être réduite au minimum.

Approche suivie. Pour atteindre les objectifs que nous nous sommes fixés, nous avons opté pour un *brain-switch* (BS) avec un **paradigme d'apprentissage adaptatif**.

Un BS est un interrupteur commandé par activation volontaire cérébrale. Il s'agit de la plus élémentaire des ICM asynchrones puisqu'elle implique seulement une seule tâche mentale spécifique à détecter (détection binaire) dans une activité EEG de fond. Dans une approche d'ICM robuste, le BS est un composant essentiel.

De plus, la phase de calibrage et d'apprentissage des algorithmes et du sujet se déroulent conjointement. Pour mener à bien la tâche de détection, l'algorithme doit apprendre les caractéristiques de l'activité spécifique afin de pouvoir la discriminer par la suite. De la même manière, l'utilisateur doit apprendre à moduler son activité cérébrale pour produire une activité suffisamment spécifique et reproductible pour être discriminée de l'activité de base du cerveau. On nomme ce processus conjoint la **co-adaptation**.

Usuellement, l'apprentissage des algorithmes et celui de l'utilisateur sont réalisés séquentiellement [88]. En général, on commence d'abord par entraîner les algorithmes, il faut donc leur fournir des données. On effectue donc un premier enregistrement, où l'utilisateur réalise un certain nombre de fois une tâche mentale prédéterminée. Étant donné que les algorithmes ne sont pas encore entraînés, l'utilisateur n'a aucun moyen de savoir s'il réalise correctement sa tâche mentale. À l'issue de cette étape, on peut entraîner les algorithmes à détecter l'activité spécifique. On relance ensuite un second enregistrement, similaire au premier, mais en affichant un signal de retour (*feedback*) c'est-à-dire le résultat de la détection de l'algorithme afin que l'utilisateur puisse apprendre à réaliser de manière reproductible sa tâche mentale. L'introduction du *feedback* modifie généralement [22] la manière dont l'utilisateur réalise sa tâche mentale, on alterne donc régulièrement les phases d'apprentissage de l'algorithme et de l'utilisateur comme l'illustre la figure 7.4

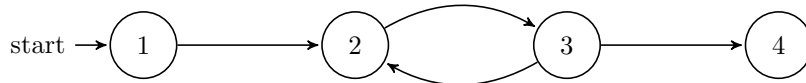


FIGURE 7.4 – Processus d'apprentissage Algorithmes/Utilisateur. **Étape 1** : Enregistrement des données sans *feedback*, **Étape 2** : Entraînement des algorithmes, **Étape 3** : Enregistrement des données avec *feedback*, **Étape 4** : Utilisateur et algorithmes entraînés

Dans notre cas, l'approche choisie est différente, l'apprentissage des algorithmes et de l'utilisateur se fait au cours de l'expérimentation, en une seule étape. Cette approche a de nombreux avantages. En premier lieu, elle permet d'accélérer la phase d'apprentissage nécessaire à l'utilisation de l'ICM. En effet elle raccourcit l'étape 1 qui consiste à enregistrer des données sans *feedback* mais elle permet aussi d'augmenter l'interaction entre l'apprentissage des algorithmes et l'apprentissage de l'utilisateur, ce qui réduit considérablement le temps pour atteindre l'étape 4, où l'utilisateur et la machine sont parfaitement entraînés.

Ainsi, l'utilisateur peut améliorer la manière dont il effectue sa tâche mentale et observer quasiment instantanément l'effet qui en résulte. Cela permet aussi d'impliquer plus fortement l'utilisateur dans le processus de calibrage de l'ICM, ce qui a pour effet d'augmenter son attention et ainsi conserver un niveau élevé de qualité des signaux. De plus, plutôt que d'utiliser des phénomènes physiologiques (rebond Beta [94], basses fréquences [21]), on utilisera la capacité de l'utilisateur à apprendre à générer une activité spécifique, quelle qu'elle soit. Ainsi, l'activité spécifique utilisée n'est pas définie à l'avance et peut évoluer au cours de l'apprentissage.

En revanche, cette approche introduit plusieurs inconvénients. Premièrement, l'implémentation des algorithmes devient plus complexe et peut conduire à des situations instables. Si le délai de réaction des algorithmes est trop lent, l'utilisateur qui adapte sa tâche mentale ne pourra pas en voir l'effet et sera tenté de la modifier de manière encore plus significative, ce qui ne fera qu'introduire de la variabilité supplémentaire et conduira à de moins bonnes performances. De la même manière, il faut aussi introduire un facteur d'oubli afin que les données en début de test ne viennent pas perturber les algorithmes en introduisant de plus en plus de variabilité. De plus, cet apprentissage peut être assez fatiguant pour l'utilisateur qui devient actif et ne se contente plus de répéter simplement sa tâche mentale pour fournir des données d'apprentissage aux algorithmes. Lorsque ce processus converge rapidement il est très efficace mais dans le cas où des difficultés sont rencontrées, il peut devenir très frustrant. Enfin, il introduit des contraintes fortes au niveau de l'exécution en temps réel puisque le temps nécessaire à la mise à jour des algorithmes doit être strictement inférieur au débit des données. De plus, se pose la question de la gestion des artéfacts qui ne peut plus se faire a posteriori.

7.3 Brain-switch : état de l'art

On appelle *brain-switch* (BS) un type particulier d'ICM qui remplit une fonction d'interrupteur. Plus particulièrement, un BS est une ICM asynchrone assurant une tâche de détection d'une activité cérébrale spécifique parmi un flux continu d'activité cérébrale non spécifique. Un BS possède principalement deux usages :

1. Déclencher la réalisation d'une action élémentaire, par exemple allumer une lumière ou fermer une main robotisée [87].
2. Signifier à la machine que l'utilisateur souhaite interagir avec elle. Dans ce cas, le BS est bien souvent couplé avec un autre type d'ICM et il est un préalable à son enclenchement. On parle alors d'ICM hybride [90]. On pourra par exemple utiliser un BS pour activer ou désactiver une ICM synchrone fondée sur des potentiels évoqués SSVEP, cette dernière permettant de commander les différents mouvements d'une orthèse [95].

Historiquement introduit par Birch et Mason [80] en 2000 sous le nom de Low Frequency Asynchronous Switch Design (LF-ASD) car fondé sur des caractéristiques basse-fréquence (dans la bande 1-4 Hz), le BS a suivi un certain nombre de développements de la part de la même équipe les années suivantes [17, 133, 20]. Le développement du BS observe un regain d'activité à la fin des années 2000 quand il est repris par l'équipe du professeur Pfurtscheller [94, 95, 95, 88]. Ne permettant de réaliser qu'une seule action, le BS semble moins attractif que les ICMs synchrones à un ou deux degrés de liberté, ce qui explique le relatif désintérêt qu'il a suscité au milieu de la précédente décennie. En réalité, le BS est le premier pas vers une ICM utilisable dans la vie réelle car il traite de manière élémentaire la problématique des interfaces purement asynchrones. Ainsi il fait aujourd'hui l'objet de plus en plus de réalisations.

Minimisation du taux de faux positif. Comme tout autre type d'ICM, le nombre de faux positif (FP - déclenchement involontaire de l'interrupteur) doit être minimisé. En particulier dans le cadre des ICMs hybrides, le nombre de FP est le facteur dominant dans l'optimisation des paramètres de l'ICM. Il doit être maintenu le plus faible possible, bien souvent au détriment de la réactivité de l'ICM. Ainsi dans [27] la spectroscopie proche infrarouge est utilisée pour réaliser un BS à partir de la réponse hémodynamique consécutive à une tâche d'imagerie motrice. Cette modalité étant peu sensible aux perturbations électromagnétiques ou aux artéfacts musculaires, la robustesse du BS s'en trouve grandement améliorée. Dans [103], un taux de FP inférieur à 1% a été obtenu en demandant

au sujet de répéter consécutivement des mouvements imaginés et en moyennant les réponses ERD/ERS correspondantes. Un seuil élevé est calculé à partir d'une courbe ROC afin de garantir un nombre de FP très faible. En contrepartie, le temps nécessaire à l'activation du BS est relativement long, entre 10 et 20 secondes. L'équipe de G.E. Birch considère que le taux de FP ne doit pas excéder 1% si l'on souhaite utiliser un BS dans une application réelle. Ce but est atteint dans leurs derniers développements grâce à une procédure complexe comprenant de la sélection de caractéristiques et une combinaison de classifieur SVM, dont les paramètres sont optimisés par un algorithme génétique hybride [34]. En revanche, un temps d'activation trop long peut être incompatible avec certaines applications, il y a donc un compromis à trouver entre ce taux maximal de FP et les difficultés d'activation du BS.

Principes physiologiques utilisés. Réaliser un BS c'est avant tout être capable de détecter de manière asynchrone l'occurrence d'une activité spécifique parmi un flux continu de signaux EEG. Contrairement aux autres types d'ICM, on ne dispose pas d'une indication temporelle (même approximative) sur le commencement de la tâche mentale. De même, il est généralement plus facile de discriminer une tâche mentale caractéristique par rapport à une autre que de la discriminer par rapport à une activité non spécifique. Il faut donc choisir une tâche mentale qui soit reconnaissable par rapport à une activité quelconque. De plus, il est nécessaire que l'activité générée soit indépendante de la plupart des autres activités. Ainsi les principales réalisations de BS se basent sur des principes physiologiques bien connus et particulièrement caractéristiques. Dans [80], la tâche mentale utilisée est générée par l'intermédiaire d'une flexion particulière, et peu usuelle, de l'index. Un jeu de caractéristiques est extrait à partir de la bande 1-4 Hz des signaux issus de 6 paires d'électrodes placées entre le cortex frontal et le cortex moteur. Dans [94], c'est le rebond (augmentation de puissance) de puissance dans la bande Beta consécutif à une flexion imaginée des deux pieds qui est utilisé pour déclencher l'interrupteur.

Une mise en œuvre simple. Un BS est en général un système simple à mettre en œuvre. Dans la majorité des cas, le nombre d'électrodes est inférieur à la dizaine. Quand la détection se base sur le rebond Beta ou les ERD/ERS, un simple filtrage spatial laplacien est utilisé pour obtenir un seul signal à partir des 5 électrodes placées au dessus de la zone motrice visée [88]. Il est ainsi aisé de coupler le BS avec un autre type de BCI ou une autre modalité. L'apprentissage est également relativement court car il est nécessaire de ne caractériser qu'une seule tâche mentale.

Exemple de réalisations. Les exemples de commande d'actions élémentaires sont peu nombreux, car disposer d'une seule commande est généralement trop limitant pour nombre d'applications. Dans [87], la détection asynchrone de l'imagination d'un mouvement de la main gauche déclenche l'ouverture ou la fermeture d'une main artificielle. Dans [86], le problème du nombre de commandes disponibles est contourné en codant dans la durée d'activation du BS deux commandes différentes, à la manière d'un signal PWM (Pulse-width modulation). A partir du signal issu d'un groupe d'électrode (avec filtrage spatial Laplacien) au-dessus d'une zone motrice (C3, C4 ou Cz), l'utilisateur peut contrôler à la fois l'ouverture/fermeture d'une main en réalisant une tâche d'imagerie motrice pendant un temps court ($2 \leq t \leq 5$ s), et la flexion extension d'un bras en réalisant la même tâche pendant un temps long ($t \geq 5$ s). Dans le cadre des ICM hybrides, le but principal est de minimiser le taux de FP en désactivant l'ICM la moins robuste quand l'utilisateur n'a pas l'intention de s'en servir. Ainsi dans [95] un BS fondé sur de l'imagerie motrice permet d'enclencher une ICM de type SSVEP pour contrôler une orthèse. L'utilisateur souhaitant initier un mouvement de l'orthèse enclenche une première fois le BS, il va ensuite choisir le type de mouvement en portant son regard sur une des deux LED placées sur l'orthèse et

clignotant à des fréquences de 8 et 13 Hz. Une fois le mouvement terminé, il enclenchera une fois encore le BS ce qui désactivera l'ICM par SSVEP. Il est ainsi montré que pendant les périodes de repos où l'utilisateur n'a pas d'intention de contrôle, le nombre de FP est passé de 5.4/minutes à 1.46/minutes grâce à l'introduction du BS. Une expérience similaire où le BS est réalisé par NIRS est également présenté dans [90].

Campagne I : Validation de l'apprentissage adaptatif

8.1 Paradigme

Le paradigme de notre première expérimentation BS est le suivant. L'utilisateur dispose d'une IHM composée principalement de l'affichage sur un écran d'ordinateur d'une barre verticale et d'un seuil comme l'illustre la figure 8.1. L'utilisateur contrôle la hauteur de la barre verticale, son but étant de lui faire dépasser le seuil pendant un certain temps, ce qui aura pour effet de déclencher l'activation du BS. Lorsque l'algorithme détecte la réalisation de la tâche mentale, la hauteur de la barre augmente et inversement, la fait baisser lorsqu'il ne détecte rien. Ainsi, l'utilisateur doit effectuer sa tâche mentale suffisamment longtemps pour que la barre monte jusqu'à dépasser le seuil. On est donc en présence d'une commande à mémoire temporelle.

Pour les besoins d'apprentissage des algorithmes qui sont majoritairement **supervisés**, on doit savoir quand l'utilisateur est en train d'effectuer sa tâche mentale. Puisque que l'on souhaite un paradigme quasiment identique que l'on soit pendant la phase d'apprentissage ou en phase d'utilisation, il n'est pas envisageable que cela soit l'ordinateur qui dicte quand l'utilisateur doit réaliser sa tâche mentale comme c'est le cas pour les paradigmes *cue-based*. Le choix s'est naturellement porté vers un paradigme asynchrone (*self-paced*). L'utilisateur signifie donc à la machine le début de sa tâche mentale en appuyant sur une touche du clavier. L'utilisateur garde ainsi le contrôle sur le timing de son apprentissage et le passage à l'utilisation réelle est transparente, l'utilisateur n'a simplement plus besoin d'appuyer sur le clavier. En revanche, cela peut introduire des artefacts et des tâches mentales parasites et concurrentes à la tâche mentale principale dans les signaux.

8.2 Chaîne de Traitement

Les algorithmes utilisés pour cette première chaîne sont des algorithmes classiques utilisés en BCI. Le développement de cette chaîne de traitement a été amorcée en début de thèse, parallèlement aux développements d'algorithmes originaux. Les expérimentations présentées dans ce chapitre ont été réalisées, pour une grande majorité d'entre elles, avant le développement des méthodes utilisant la géométrie Riemannienne.

La chaîne de traitement est donc constituée d'algorithmes standards qui ont été implémentés et modifiés pour être rendus adaptatifs. La figure 8.2 schématise les différents blocs utilisés, dont 3 sont adaptatifs : la détection d'artéfact, le filtrage spatial par CSP et la classification par LDA.

Filtrage fréquentiel. La fréquence d'échantillonnage des signaux est de 512 Hz. Le filtrage fréquentiel est effectué par l'intermédiaire d'un filtre IIR de Butterworth d'ordre 4 entre 8 et 30 Hz. Si l'utilisateur a déjà effectué une expérience, il peut être spécifique à la tâche mentale qu'il utilise. Le filtrage entre 8 et 30 Hz reste valide pour un grand nombre de tâches mentales, en particulier les tâches mentales d'imagination motrice qui activent

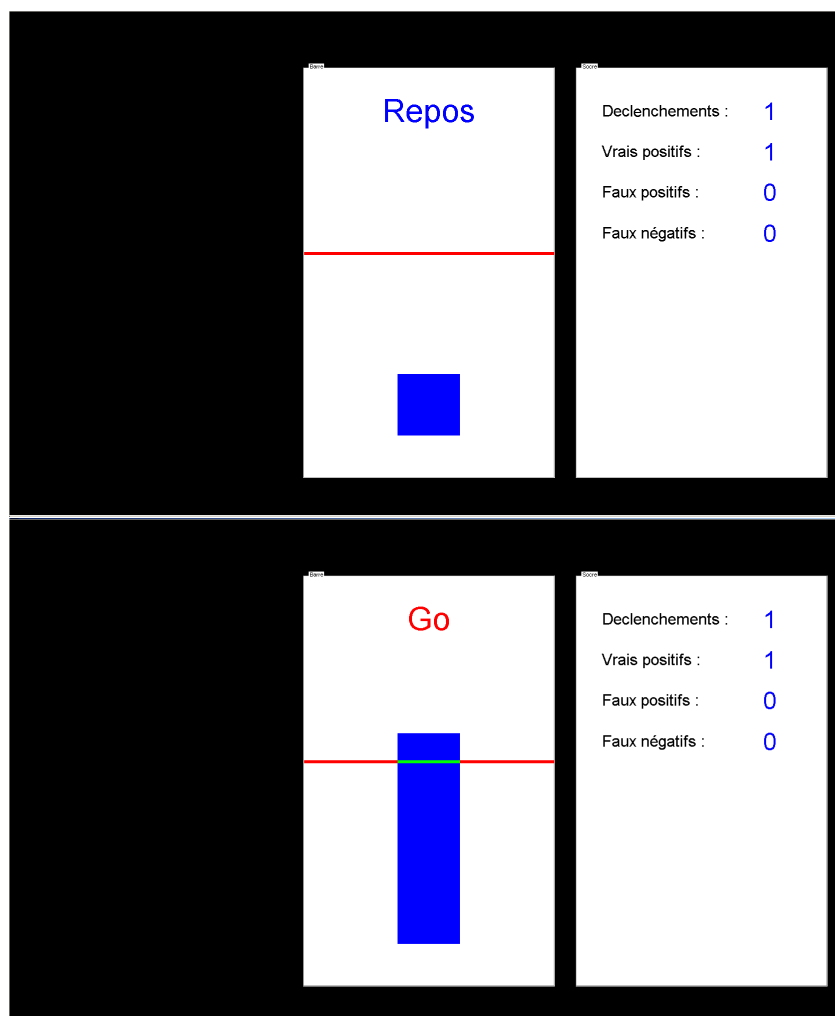


FIGURE 8.1 – IHM présentée à l'utilisateur du Brain-Switch

principalement les bandes Alpha et Beta (cf section 2.2.2). Il permet aussi de s'affranchir d'une grande partie des artéfacts oculaires.

Estimation des matrices de covariance. Ce bloc fournit les données de base à toute la chaîne de traitement, il permet aussi de donner la cadence à l'ensemble de la chaîne de traitement. Une matrice de covariance est calculée sur 128 échantillons, soit tous les 250 ms. Cette matrice est ensuite envoyée aux différents blocs pour y être classée. Ainsi le système de détection prend 4 décisions par seconde, ce qui permet d'avoir un système plutôt réactif.

Détection d'artéfact. Chaque matrice de covariance passe par la détection d'artéfact pour être avalisée. Si le bloc de détection d'artéfact décide que la matrice courante contient des données anormales, il le signalera à l'ensemble des blocs successifs afin que celle-ci soit ignorée. La détection d'artéfact est adaptative et supervisée. Elle se fonde sur un critère de dissimilarité du déterminant des matrices de covariance. On calcule ce critère pour chaque classe, afin d'éviter qu'une matrice appartenant à une classe, étant par nature dissimilaire de l'ensemble des matrices de l'autre classe, ne soit considérée comme un artéfact.

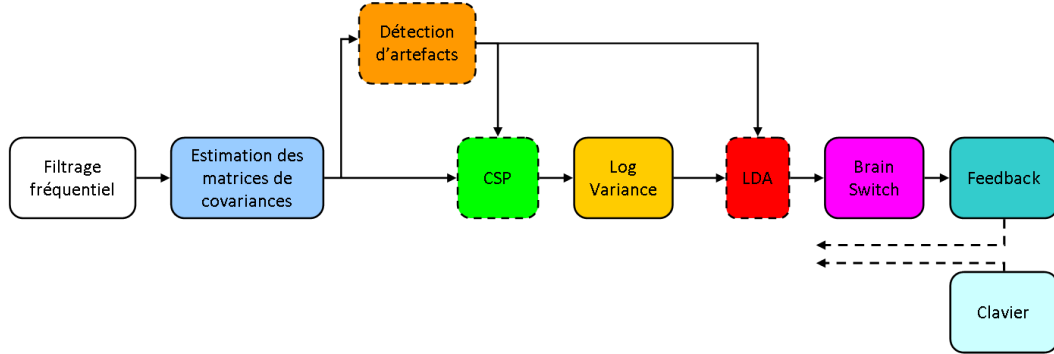


FIGURE 8.2 – chaîne de traitement du Brain Switch

Soit Σ_i la matrice de covariance estimé sur la fenêtre d'indice i et de classe k , la détection d'artéfact se fonde sur le déterminant de cette matrice $d_i = \det(\Sigma_i)$. On calcule ensuite une distance normalisée (distance de Mahalanobis à 1 dimension) sur cette variable :

$$\tilde{d}_i = \sqrt{\frac{(d_i - \mu_i^{(k)})^2}{(\sigma_i^{(k)})^2}}$$

avec $\mu_i^{(k)}$ et $\sigma_i^{(k)}$ respectivement la moyenne et l'écart type à l'instant i des déterminants d_i des matrices de covariance de classe k . Ces deux paramètres sont mis à jour à chaque nouvelle matrice de covariance :

$$\mu_i^{(k)} = \frac{1}{i} \sum_{j=1}^i d_j = \frac{i-1}{i} \mu_{i-1}^{(k)} + \frac{1}{i} d_i$$

et

$$\sigma_i^{(k)} = \left[\frac{1}{i} \sum_{j=1}^i (d_j - \mu_i^{(k)})^2 \right]^{\frac{1}{2}} \approx \left[\frac{i-1}{i} \sigma_{i-1}^{(k)} + \frac{1}{i} (d_i - \mu_i^{(k)})^2 \right]^{\frac{1}{2}}$$

La condition de rejet est alors simplement :

$$\tilde{d}_i > \kappa$$

avec κ un seuil placé généralement à 2.5 , ce qui correspond à un déterminant qui dévie de plus de 2.5 fois l'écart-type des données.

Si la matrice est rejetée, les paramètres i , $\mu_i^{(k)}$ et $\sigma_i^{(k)}$ seront réinitialisés à la valeur précédente ($i = i - 1$).

Ce critère de rejet est directement inspiré de la méthode MCD (*Minimum Covariance Déterminant*, cf annexe A) pour le calcul de la matrice de covariance, qui dit implicitement que plus une matrice contient d'artefact, plus son déterminant est élevé. On considère aussi qu'au sein d'une même classe, le déterminant des matrices de covariance est relativement semblable. Cette approche originale donne de bons résultats en pratique pour un coût de calcul faible.

Filtrage spatial par CSP. Ce bloc est décisif : il permet d'extraire l'information spatiale afin de la présenter à l'algorithme de classification. Le côté adaptatif de cet algorithme est traité par la mise à jour des matrices de covariance moyennes par classe. A titre de rappel les

filtres spatiaux par **CSP** se calculent via la diagonalisation conjointe des deux matrices de covariance moyennes de chaque classe $\Sigma^{(1)}$ et $\Sigma^{(2)}$ (cf section 3.2.3). On introduit également un facteur d'oubli τ dans le but de ne plus prendre en compte les matrices de covariance estimée trop loin dans le passé, vraisemblablement peu représentatives de la tâche mentale actuelle. Soit $a_i^{(k)}$ le scalaire ayant pour valeur 1 si la matrice de covariance Σ_i ne contient pas d'artéfact et est de classe k , et valant 0 sinon. Soit le vecteur $\mathbf{a}_i^{(k)} = [a_i^{(k)} \dots a_{i-\tau}^{(k)}]$. On met à jour les matrices de covariance moyenne par la formule de la moyenne arithmétique :

$$\begin{aligned}\Sigma_i^{(k)} &= \frac{1}{\|\mathbf{a}_i^{(k)}\|_0} \sum_{j=i-\tau}^i a_j^{(k)} \Sigma_j \\ &= \frac{1}{\|\mathbf{a}_i^{(k)}\|_0} \left[\|\mathbf{a}_{i-1}^{(k)}\|_0 \Sigma_{i-1}^{(k)} + a_i^{(k)} \Sigma_i - a_{i-\tau-1}^{(k)} \Sigma_{i-\tau-1} \right]\end{aligned}\quad (8.1)$$

Avant d'effectuer cette mise à jour, on s'assurera que $\|\mathbf{a}_i^{(k)}\|_0 \neq 0$. Dans les expérimentations conduites dans notre laboratoire, ce facteur d'oubli a été fixé à 1500 points, ce qui correspond à environ 6 minutes d'enregistrement. On calcule ensuite les filtres spatiaux \mathbf{W}_i par diagonalisation jointe des deux matrices mises à jour. Le problème qui apparaît est que d'une itération à l'autre, les filtres spatiaux peuvent permuter au sein de \mathbf{W}_i , voire même changer de signe. Il est nécessaire qu'un filtre garde sa place et son signe au niveau de la matrice. Il faut donc les repositionner en fonction de la place qu'ils occupaient à l'étape précédente.

En considérant qu'une itération modifie peu la topologie des filtres spatiaux, on introduit une mesure de distance entre les filtres spatiaux. Avec $\mathbf{W}_i = [\mathbf{w}_{1,i} \dots \mathbf{w}_{c,i} \dots \mathbf{w}_{C,i}]$ on a :

$$d(\mathbf{w}_{c,i}, \mathbf{w}_{c,i-1}) = \left\| \left(\frac{\mathbf{w}_{c,i}}{\|\mathbf{w}_{c,i}\|} \right)^2 - \left(\frac{\mathbf{w}_{c,i-1}}{\|\mathbf{w}_{c,i-1}\|} \right)^2 \right\|$$

avec ici l'exposant ² comme le carré individuel de chaque élément du vecteur. Cette distance est invariante par changement de signe et par changement d'échelle. On calcule ainsi pour chaque couple de filtre la distance correspondant ce qui permet de retrouver les filtres ayant changé de place et d'effectuer la permutation et le changement de signe au besoin.

Toutes ces opérations sont relativement délicates, mais elle fonctionnent bien pour les filtres spatiaux les plus discriminants car ils sont très caractéristiques et changent peu d'une itération à l'autre.

Rétrospectivement, la mise à jour des filtres via la diagonalisation jointe des matrices de covariance moyennes n'était pas la solution optimale. Une solution plus élégante et n'ayant pas ce problème de permutation repose sur la mise à jour directe des filtres à partir d'un calcul itératif [132] :

$$\mathbf{w}_{c,i} = \frac{\mathbf{w}_{c,i-1}^T \Sigma_i^{(1)} \mathbf{w}_{c,i-1}}{\mathbf{w}_{c,i-1}^T \Sigma_i^{(2)} \mathbf{w}_{c,i-1}} (\Sigma_i^{(1)})^{-1} \Sigma_i^{(2)} \mathbf{w}_{c,i-1} \quad (8.2)$$

Extraction des caractéristiques : Log variance. Ce simple bloc effectue la projection de la matrice de covariance par les filtres spatiaux, conserve uniquement la diagonale et en prend le logarithme :

$$\mathbf{v}_i = \log(\text{diag}(\mathbf{W}_i^T \Sigma_i \mathbf{W}_i))$$

On obtient un vecteur de caractéristiques pour chaque matrice. On conserve l'ensemble des C filtres spatiaux, ce qui produit des caractéristiques de dimension C . Dans cette expérience,

nous n'effectuons pas de sélection de filtres spatiaux ($L = C$) pour ne pas ajouter une source d'invariance supplémentaire au niveau des caractéristiques. Ici $C = 16$ ce qui ne posera pas de problème de sur-apprentissage à la classification.

Classification par LDA. Une classification par LDA (section 3.3.2) est appliquée sur ce vecteur de caractéristiques. La LDA est une technique très peu demandeuse en ressources de calcul puisque sa résolution fait simplement appel à un calcul de covariance et à une décomposition en vecteurs propres. Le problème de l'adaptation a donc été simplement traité par la mise à jour de la base d'apprentissage. On apprend donc un classifieur à chaque fois qu'un nouveau vecteur de caractéristiques \mathbf{v}_i est disponible. Un facteur d'oubli est également introduit pour prendre en compte la modification des filtres spatiaux au cours du temps. Ce facteur d'oubli est réglé à 750 échantillons, ce qui correspond à 3 minutes environ. L'apprentissage du classifieur s'effectue donc sur les 3 dernières minutes d'enregistrement EEG. Évidemment, le vecteur \mathbf{v}_i que l'on classe à l'instant i n'est pas pris en compte dans la base d'apprentissage du classifieur calculé au même instant, mais le sera à l'instant $i + 1$. La sortie y_i de la LDA est une valeur binaire, 1 si on détecte la présence d'une activité spécifique et 0 sinon.

Brain-Switch. Dans la plupart des ICMs, le traitement s'arrête à cette étape de classification et on utilise généralement la sortie de la classification comme commande pour l'effecteur. Dans notre approche nous avons ajouté une étape de post-traitement. Cette étape a deux buts :

1. adapter la sortie du classifieur pour une prise de décision plus robuste sur le déclenchement du BS.
2. adapter la sortie du classifieur pour afficher un *feedback* non saccadé.

Pour atteindre ces deux buts, on effectue donc un filtrage passe-bas IIR sur la sortie du classifieur. Le bloc de filtrage IIR de Simulink permet d'obtenir une sortie continue, que l'on ré-échantillonnera à 30 Hz pour un affichage du *feedback* non saccadé. Cette nouvelle sortie \tilde{y}_i est ensuite utilisée pour prendre une décision concernant le déclenchement du BS. Si le seuil th , fixé arbitrairement à 0.8, est dépassé pendant un certain temps noté τ_d (réglé à 1 s), le déclenchement du BS est lancé. Ce processus est schématisé figure 8.3.

C'est donc dans la durée de maintien de la tâche mentale que l'utilisateur va pouvoir activer l'interrupteur et non pas dans la simple détection de l'activité spécifique. Le *feedback* est alors très important car il permet à l'utilisateur de savoir combien de temps il doit maintenir sa tâche mentale.

Enfin, une fois le BS enclenché, il faut attendre un certain temps (ici 2 s) avant que le BS se réamorce et que le prochain déclenchement soit possible. Ce délai permet d'éviter les déclenchements consécutifs trop rapides alors que l'utilisateur ne souhaite probablement activer l'interrupteur qu'une seule fois.

Feedback Le *feedback* s'effectue suivant deux modalités. On a un premier retour visuel constitué de la barre verticale et du seuil imposé qui affiche en continue la sortie de l'algorithme de classification, filtrée, rééchantillonnée et écrêtée, comme en témoigne la figure 8.3. Cette information permet à l'utilisateur d'être informé en temps réel de la bonne prise en compte de sa tâche mentale par l'algorithme, et le cas échéant, de l'ajuster. On dispose également d'un *feedback* sonore qui avertit l'utilisateur du déclenchement du BS.

Remarques générales sur la chaîne de traitement. Cette chaîne de traitement fait appel à des méthodes classiques de traitement du signal. La mise en œuvre adaptative de ces algorithmes a introduit un certain nombre de difficultés. Un travail important a été mené pour rendre cette chaîne la plus ergonomique possible de la part de l'utilisateur. Du

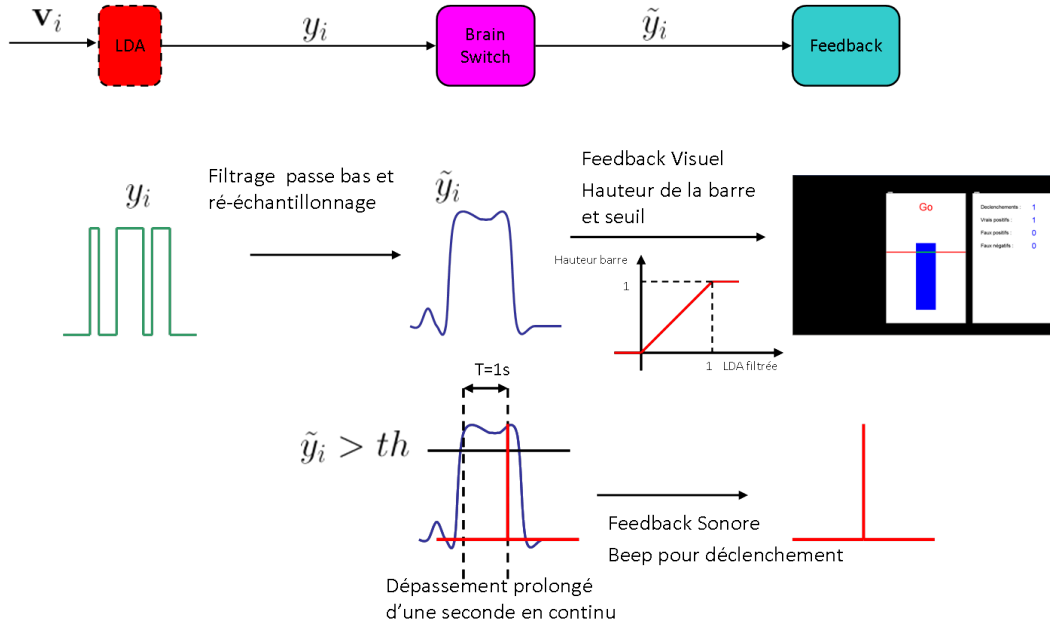


FIGURE 8.3 – Post-traitement après la classification par LDA

point de vue de l'IHM, il a fallu simuler le contrôle continu à partir d'une sortie discrète de fréquence 4 Hz. En effet il est très peu agréable de voir le retour visuel évoluer de manière saccadée, le filtrage et le ré-échantillonnage a donc ici un double emploi. Enfin les différents paramètres de la chaîne (facteur d'oubli, seuil etc) ont été réglés empiriquement lors des premières expériences de validation de la chaîne. Pour ce qui concerne les facteurs d'oubli, un compromis a été fait entre réactivité à la modification de la tâche mentale et la stabilité des algorithmes. Pour le seuil et le délai avant déclenchement, les paramètres sont le fruit d'un compromis entre robustesse et difficulté de déclenchement.

8.3 Campagne de test

8.3.1 Présentation de l'expérience.

Dans le but de valider le fonctionnement du BS, outre les nombreuses expérimentations menées lors de son développement, une première campagne de tests a été lancée. Il a été fait appel à 10 sujets volontaires, dont 7 naïfs¹, 2 ayant déjà participé à une seule session ICM d'un paradigme différent (quasiment naïf) et un utilisateur expérimenté.

La totalité des 16 voies de l'amplificateur a été utilisée. Le montage des électrodes est donné figure 8.4 et il est construit dans le but de couvrir le plus largement possible les différentes zones du cerveau, en particulier les zones motrices.

La campagne de test compte une seule session par sujet, d'une durée d'une heure environ comportant 30 minutes d'enregistrement. Une session se divise en quatre étapes comme l'illustre la figure 8.5.

Les deux premières étapes A et B sont des paradigmes classiques de type *cue based*. il s'agit d'enchaîner sans *feedback* des réalisations de tâches mentales imposées. L'étape A dure 5 minutes et l'on doit effectuer 10 réalisations des 3 tâches mentales suivantes : mouvement réel de la main droite, mouvement réel du pied et repos. L'étape B est sensiblement identique

1. Terme consacré pour un utilisateur n'ayant jamais participé à une expérience ICM

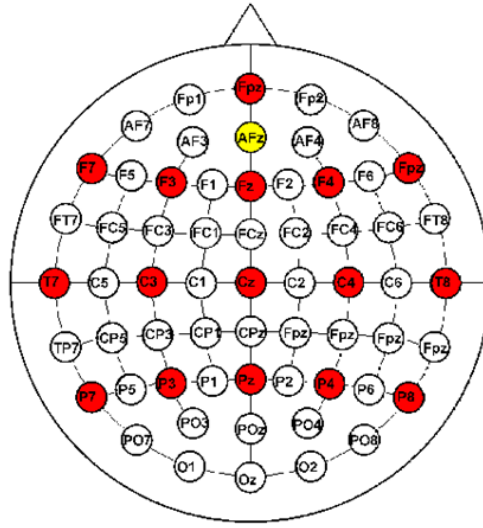


FIGURE 8.4 – Position des électrodes pour la campagne de test Brain Switch. La masse est en jaune. L'électrode de référence est placée sur l'oreille.

à la différence que les tâches mentales sont ici des mouvements imaginés. On réalise 20 tâches mentales de chaque classe, le test dure donc 10 minutes. Ces deux étapes ne sont pas indispensables pour la suite, elles permettent de familiariser le sujet naïf avec le processus de réalisation de tâche mentale qui n'est pas entièrement naturel. De plus, ces jeux de données seront utiles pour le développement ultérieur d'algorithmes. Ainsi, le jeu de données portant sur les mouvements imaginés est utilisé dans le chapitre 4.

Les étapes C et D sont le cœur de la session, le BS. L'étape C est la partie apprentissage adaptative, elle dure 15 minutes maximum, et peut être arrêtée si le sujet atteint un niveau stable de performances. Au départ de l'expérience, l'utilisateur n'est pas contraint de choisir une tâche mentale particulière mais la consigne lui est donnée d'utiliser plutôt une tâche mentale fondée sur de l'imagerie motrice, qu'il pourra modifier lors de l'apprentissage. Enfin l'étape D est une étape de test, où les paramètres issus de la phase d'apprentissage, c'est-à-dire les filtres spatiaux et le classifieur sont fixés. Le paradigme n'est donc plus adaptatif et le test représente un cas d'utilisation réel où l'utilisateur tentera de déclencher le BS lorsqu'il le souhaitera. Cette étape dure 5 minutes. Entre chaque étape, on marque une pause afin que l'utilisateur puisse se reposer.

8.3.2 Évaluation des résultats.

Dans le but d'évaluer les résultats, nous allons définir 5 critères :

Vrai Positif (VP) : un **VP** est un déclenchement du BS alors que l'utilisateur souhaitait l'activer. L'utilisateur souhaitant activer le BS le signifie par un appui clavier, puis commence sa tâche mentale. Si un déclenchement a lieu dans les 10 s suivant un appui clavier, il est considéré comme valide.

Faux Négatif (FN) : un **FN** est un non déclenchement du BS dans les 10 s suivant un appui clavier. Un FN signifie que l'utilisateur n'a pas réussi à activer l'interrupteur alors qu'il tentait de le faire.

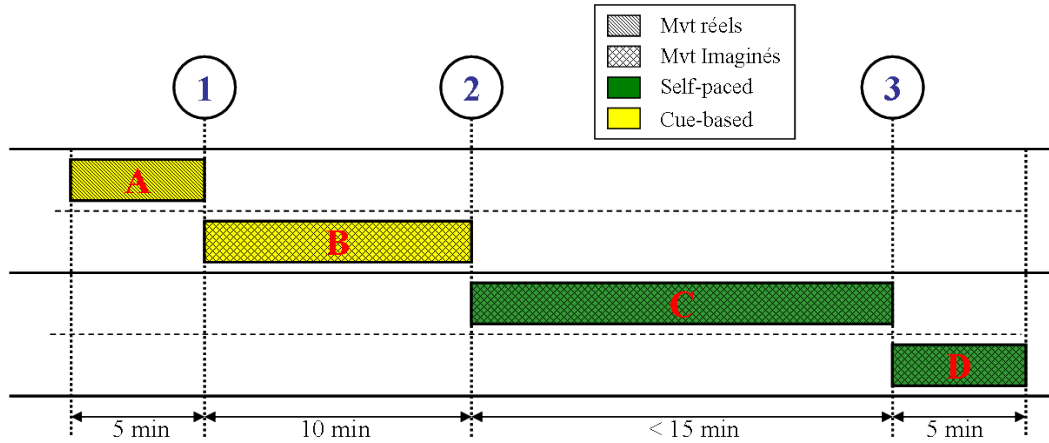


FIGURE 8.5 – Déroulement en quatre étapes d'une session de la campagne de test.

Faux Positif (FP) : un FP est un déclenchement intempestif de l'interrupteur alors que l'utilisateur ne le souhaite pas. La figure 8.6 illustre ces trois concepts dans le cadre du BS.

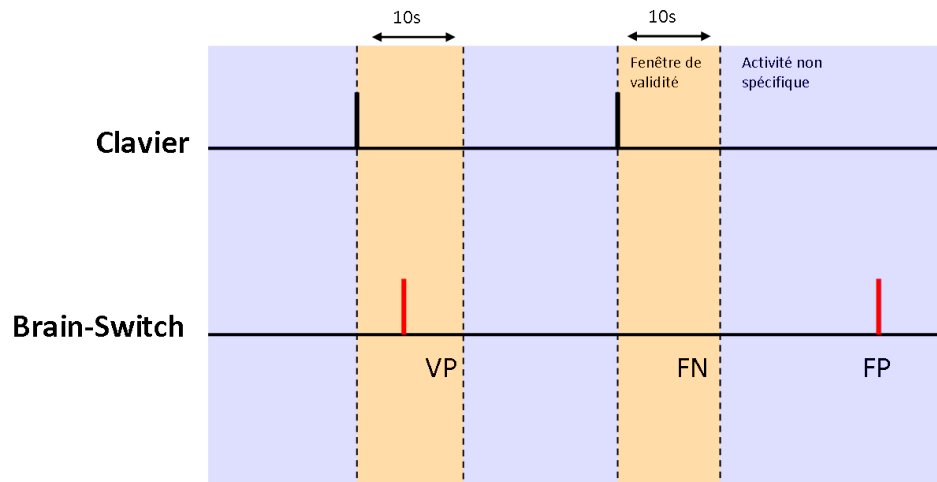


FIGURE 8.6 – Illustration des VP, FN et FP dans le cadre du Brain switch

True Positive Rate (TPR) : le TPR est une quantité représentant le ratio de tentatives de déclenchement réussies sur l'ensemble des tentatives :

$$TPR = \frac{VP}{VP + FN}$$

Ce score témoigne de la **sensibilité** du système. Un TPR de 1 signifie que l'utilisateur a réussi à activer l'interrupteur dans toutes ses tentatives.

Positive Predictive Value (PPV) : le PPV est une quantité représentant le ratio de déclenchements volontaires sur l'ensemble des déclenchements. Il se calcule par la formule :

$$PPV = \frac{VP}{VP + FP}$$

Ce score témoigne de la **sélectivité** du système. Un PPV de 1 signifie que l'ensemble des activations était voulu.

Pour un système parfait, le TPR et le PPV doivent tendre vers 1.

8.3.3 Résultats.

Nous allons dans un premier temps détailler les résultats obtenus sur un sujet avant de donner les performances globales.

Phase d'apprentissage. Pour l'utilisateur *E* l'apprentissage a duré 10 minutes environ. Durant ces 10 minutes, il a tenté de déclencher l'interrupteurs 53 fois, ce qui correspond à une tentative toutes les 12 s. La figure 8.7 montre l'évolution du nombre de VP et de FP par minute au cours de l'apprentissage. Lors des 2 premières minutes, on observe un taux de VP approximativement égal au taux de FP, ce qui signifie que l'algorithme répond de manière hasardeuse. Durant les minutes suivantes, la convergence des filtres spatiaux permet une augmentation de la sensibilité puis dans les dernières minutes, on observe une stabilisation de la classification qui fait que le nombre de FP devient nul.

Sur cette même figure, on donne un exemple de la sortie filtrée de la LDA. On note aussi des caractéristiques spatiales spécifiques au mouvement imaginé de la main, tâche mentale choisie par le sujet.

Finalement, les performances sont de 0.91 pour le TPR et 0.86 pour le PPV ce qui témoigne d'une bonne sensibilité et sélectivité.

Phase de test. Les résultats sont donnés à la figure 8.8. En observant l'évolution du nombre de VP au cours du temps, on remarque que l'utilisateur a éprouvé des difficultés en début d'enregistrement. Une fois la tâche mentale retrouvée, le test s'est déroulé correctement avec un taux approximatif de 4 VP par min. On remarque aussi qu'il n'y a eu aucun FP, ce qui dénote une bonne sensibilité du système. Enfin, le temps moyen avant déclenchement est d'environ 1.6 s après appui sur le clavier. Compte tenu du délai de maintien au dessus du seuil d'1 s nécessaire au déclenchement, le système est très réactif.

Erreur au cours de l'apprentissage. Parmi 4 des 10 sujets, un élément est venu perturber le déroulement, soit de l'apprentissage, soit du test. Dans la plupart des cas, l'élément perturbateur est de type expérimental, comme par exemple une électrode qui se décolle durant la pause entre l'apprentissage et le test et qui fait que les paramètres appris ne sont plus valides. Observons maintenant le cas d'un utilisateur ayant eu un problème au cours de l'apprentissage. Les résultats sont reportés figure 8.9. Sur la courbe montrant l'évolution du nombre de VP par min, on observe que tout se passe très bien jusqu'au milieu de l'enregistrement, avec seulement 1 FP en début d'apprentissage. En revanche, à partir du milieu de l'expérience, les performances chutent brutalement et l'utilisateur n'observe plus aucun contrôle sur le système. Au niveau des caractéristiques spatiales, on distingue clairement la présence de la caractéristique correspondant à un mouvement imaginé de la main. Cependant, en troisième position (en bas, à gauche), on note la présence d'une caractéristique spatiale dont les coefficients sont élevés pour toutes les électrodes. Ceci est typiquement le cas d'un artefact ayant touché l'ensemble des électrodes et ayant perturbé la CSP. En vérifiant les signaux a posteriori, il apparaît qu'un artefact n'a pas été détecté et s'est propagé en biaisant l'ensemble de la chaîne, qui se révèle donc être trop sensible.

Résultats globaux. Les résultats globaux sur l'étape de test sont donnés table 8.1. Si l'on prend en compte l'ensemble des utilisateurs, on observe un TPR de 0.67 et un PPV de 0.93. On peut donc dire que la chaîne globale est plutôt sélective car il y a peu de déclenchements involontaires. En revanche le TPR moyen n'est pas très élevé ce qui dénote

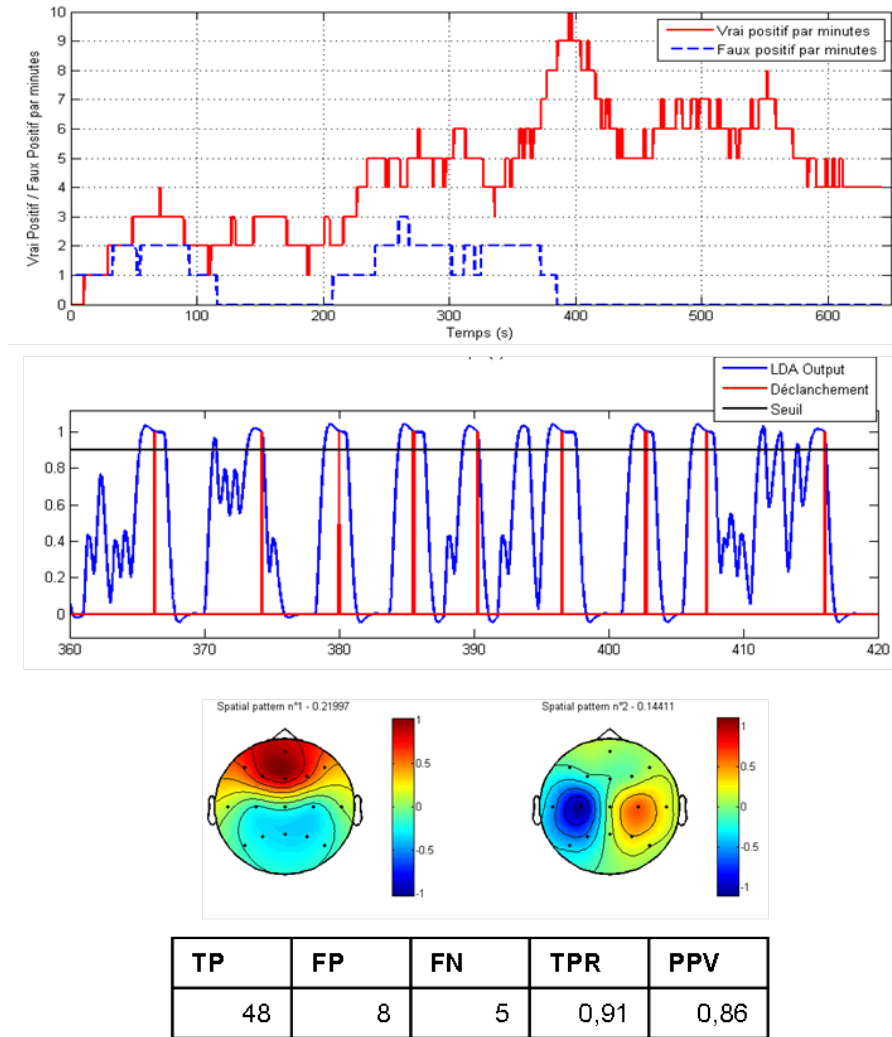


FIGURE 8.7 – Résultat de l'apprentissage pour l'utilisateur *E*. De haut en bas : évolution du nombre de VP et FP au cours du temps, exemple de feedback et de déclenchement de l'interrupteur, caractéristiques spatiales et table des performances.

une certaine difficulté à activer le BS. En réalité ce score est biaisé par les utilisateurs ayant rencontré des problèmes lors de l'expérience (noté dans le tableau par une étoile). On remarque que pour ces 4 utilisateurs, on a très peu de déclenchements (< 10). En revanche il est rassurant de voir que pour ces mêmes utilisateurs, on a un PPV très élevé ce qui signifie qu'il y a peu de déclenchements involontaires. Ainsi si la détection ne fonctionne pas (ou mal), le système ne déclenchera pas de manière intempestive l'interrupteur, ce qui pourrait avoir des conséquences dans certains cas.

En revanche, si on prend en compte uniquement les utilisateurs qui n'ont pas rencontré de problèmes lors des tests, on obtient un TPR de 0.89 et un PPV de 0.92. On a donc là de très bonnes performances qui dépassent même l'état de l'art (par exemple pour [88], on obtient un TPR de 0.79 et un PPV de 0.84). On note aussi une moyenne de 26 VP au cours de l'expérience qui a duré 5 minutes soit environ 5.2 VP par minute (soit le double des résultats obtenus dans [88]).

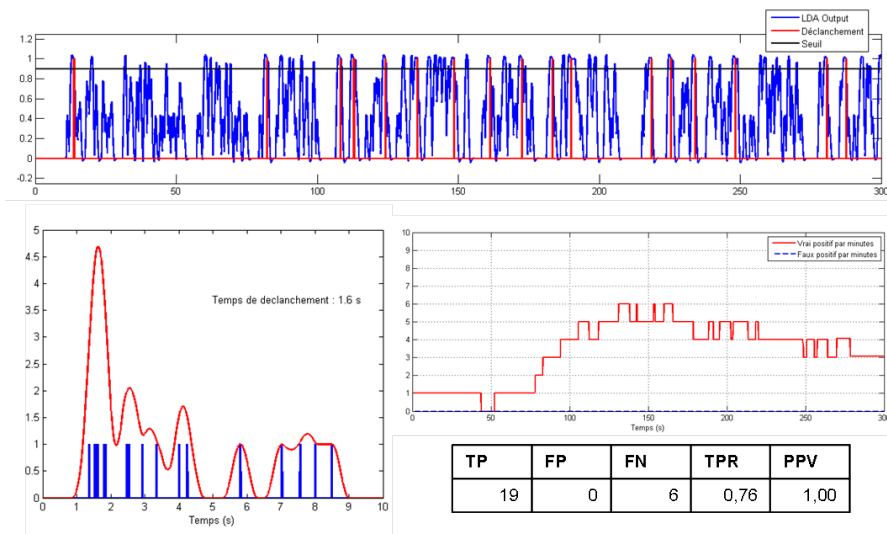


FIGURE 8.8 – Résultat du test pour l'utilisateur *E*. En haut : *feedback* et déclenchement du BS. En bas à gauche : distribution des temps de déclenchement. En bas à droite : évolution du nombre de VP et de FP au cours du temps et table de performances.

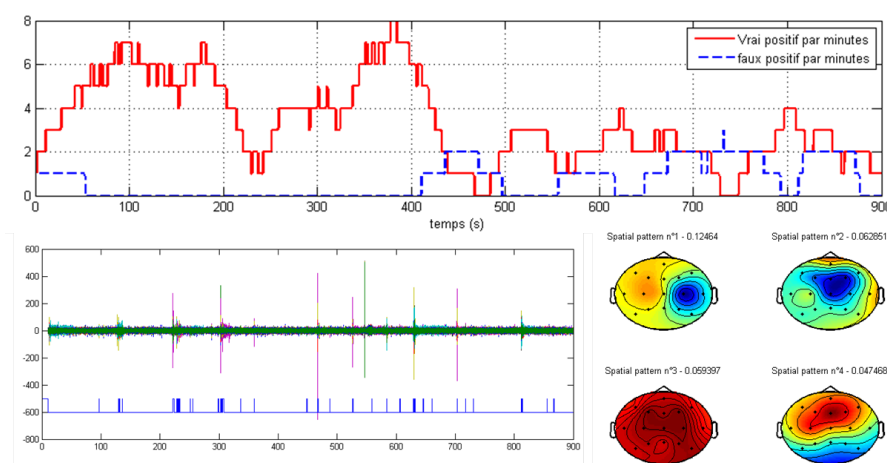


FIGURE 8.9 – Résultat de l'apprentissage pour l'utilisateur *K*. En haut : évolution du nombre de VP et de FP au cours du temps. En bas à gauche : Signaux EEG et sortie du bloc de détection d'artefact. En bas à droite : caractéristiques spatiales.

8.4 Conclusion

Ce chapitre a décrit la mise en place expérimentale d'une version préliminaire du BS. La chaîne présentée est adaptative ce qui permet un apprentissage très rapide. Ainsi en 15 minutes maximum, un utilisateur est capable de contrôler un BS avec un niveau de performance comparable à l'état de l'art. Le système est relativement robuste dans le sens où il y a peu de déclenchements involontaires de l'interrupteur. On peut également régler le niveau de robustesse en ajustant le seuil et le temps avant déclenchement. En revanche, cela se fera au détriment du taux de réussite car la condition de déclenchement sera plus dure à atteindre. Cela dépendra donc du contexte dans lequel sera utilisé le BS.

Sujet	TP	FP	FN	TPR	PPV
<i>A</i>	29	1	0	1	0.97
<i>B</i>	33	7	3	0.92	0.83
<i>E</i>	19	0	6	0.76	1
<i>G</i>	27	1	4	0.87	0.96
<i>H</i> *	1	0	11	0.08	1
<i>I</i>	18	6	4	0.82	0.75
<i>J</i> *	9	0	9	0.5	1
<i>K</i> *	8	2	10	0.44	0.8
<i>L</i> *	4	0	9	0.31	1
<i>M</i>	31	0	1	0.97	1
moyenne	17.9	1.7	5.7	0.67	0.93
moyenne *	5.5	0.5	9.5	0.33	0.95
moyenne $\bar{*}$	26	2.5	3	0.89	0.92

TABLE 8.1 – Résultat sur l'ensemble des utilisateurs

Coté utilisateur, la possibilité d'ajuster sa tâche mentale au fil de l'expérience a été bien accueillie. L'utilisateur se sent plus impliqué dans le processus d'apprentissage. Au fil de l'expérience, la tâche mentale devient de plus en plus naturelle et il est fascinant d'observer la vitesse à laquelle ce processus se produit.

Finalement, les résultats de cette première campagne de test sont encourageants et confirment la faisabilité d'une ICM orientée vers l'utilisateur, dans le cadre d'une utilisation peu contraignante, avec une mise en situation opérationnelle rapide.

Campagne II : Couplage avec un effecteur

9.1 Paradigme

En se basant sur les résultats de la première campagne de test, nous souhaitions coupler le BS ainsi développé avec un effecteur.

L'effecteur choisi est un dispositif de contrôle électronique d'environnement destiné à des personnes handicapées. Celui-ci est commercialisé par la société PROTEOR¹. Ce dispositif, nommé KÉO (en photo figure 9.1), permet de contrôler divers appareils électroniques, comme un curseur sur un ordinateur, une télévision, un téléphone ou encore un environnement domotique à l'aide d'un simple contacteur.



FIGURE 9.1 – Dispositif d'aide électronique à la personne KÉO

En effet, certaines personnes à mobilité très réduite ne peuvent utiliser qu'un contacteur, dans ce contexte, le KÉO leur permet de sélectionner une action qui leur est présentée via un mode de défilement. Notre BS étant en substance un contacteur virtuel, il peut aisément s'interfacer avec le KÉO. Le couplage du BS avec ce dispositif permet ainsi de réaliser une application répondant à un besoin réel.

Principe de fonctionnement du KÉO. Le KÉO se présente sous la forme d'un écran affichant une matrice d'actions à réaliser. La matrice est entièrement paramétrable et peut comporter 9 (3×3) ou 20 (4×5) actions différentes. Les actions sont également paramétrables et peuvent déboucher soit sur la réalisation d'une action (via l'envoi d'un code infrarouge

1. www.proteor.fr

à un dispositif externe), soit sur l'ouverture d'un sous-menu (présentation d'une nouvelle matrice d'action) sur le KÉO. Sur la figure 9.1, le dispositif présente une matrice 3×3 permettant, entre autres, d'allumer la télé, monter ou baisser le volume, allumer une lumière ou encore de composer un numéro de téléphone.

Dans notre scénario, l'utilisateur dispose d'un unique contacteur pour naviguer dans le menu et sélectionner l'action qu'il souhaite réaliser. Il devra activer successivement plusieurs fois l'interrupteur pour parvenir à déclencher une action. Ce processus s'effectue en deux phases :

1. **Asynchrone** Après une période d'inactivité, le KEO se place automatiquement en veille. Un premier déclenchement permet de sortir le dispositif de l'état de veille. Il est alors nécessaire de réaliser un deuxième déclenchement permettant de lancer un mode de défilement.
2. **Synchrone** Une fois le défilement activé, les touches vont être placées successivement en surbrillance. L'utilisateur sélectionne ainsi une action en déclenchant l'interrupteur lorsque la touche souhaitée est en surbrillance. Il y a deux modes de défilement, un défilement de type ligne/colonne (comme pour le P300 speller) et un défilement de type touche.

La figure 9.2 illustre ce processus dans le cadre d'un défilement ligne/colonne. Le défilement commence par les lignes, en passant à la ligne suivante au bout d'un temps $t > T_{def}$, paramétrable entre 0.1 et 5 secondes. Une fois la ligne sélectionnée, le défilement se fait touche par touche sur la ligne en question. Lorsque le KEO est en veille, il faut donc 4 déclenchements successifs pour lancer une action.

La figure 9.3 illustre ce processus dans le cadre d'un défilement touche par touche. Lorsque le KEO est en veille, il faut 3 déclenchements successifs pour lancer une action.

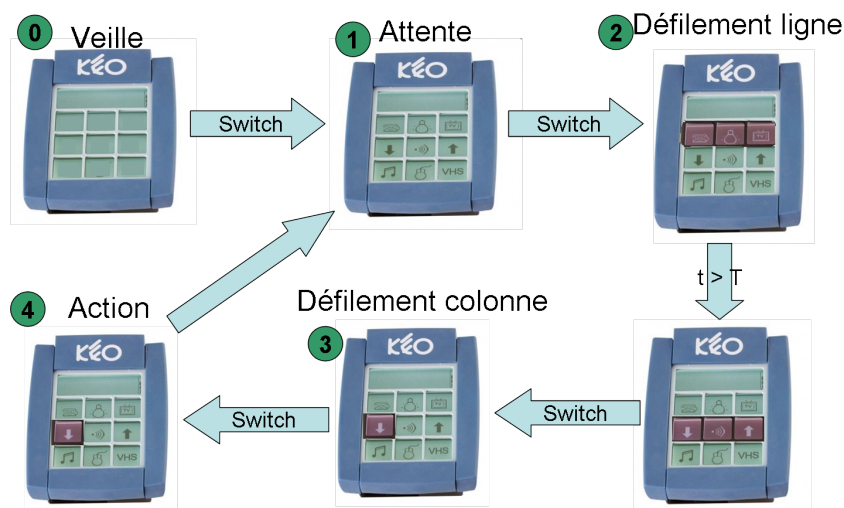


FIGURE 9.2 – Processus menant à la réalisation d'une action dans le cadre d'un défilement Ligne/Colone.

Le défilement ligne/colonne permet d'adresser rapidement un nombre important d'actions. En revanche, il n'est pas très intuitif d'un premier abord. Nous avons ainsi choisi d'utiliser un défilement touche par touche, qui est plus lent pour l'utilisateur, mais aussi plus simple. De plus les tests préliminaires ont montré un meilleur taux de réussite sur les défilements touche par touche car seulement 3 déclenchements sont nécessaires pour choisir une action. En effet, en mode défilement, l'utilisateur est contraint par le temps et doit

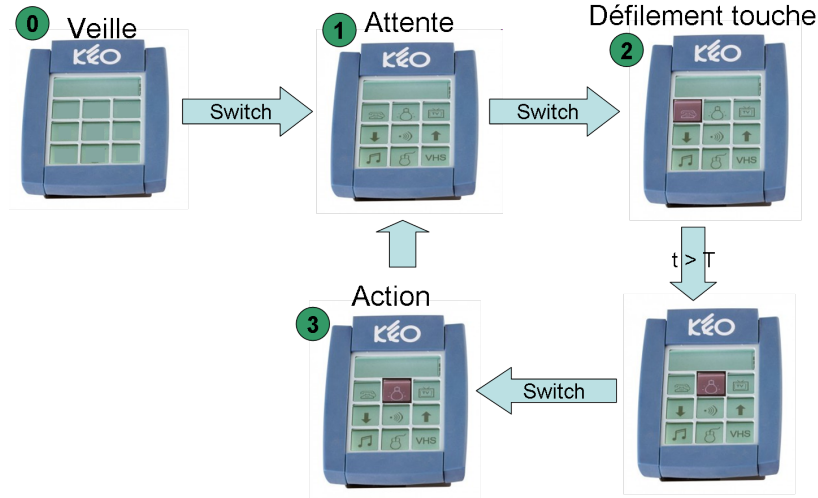


FIGURE 9.3 – Processus menant à la réalisation d’une action dans le cadre d’un défilement Touche.

déclencher l’interrupteur au bon moment. On observe donc un taux de réussite plus faible dans ce contexte.

Puisque l’utilisateur devra garder les yeux sur le KEO pour contrôler le défilement, il n’est pas envisageable de conserver le *feedback* du BS sur l’écran de l’ordinateur comme c’était le cas dans la première campagne de test.

Ainsi, un dispositif constitué d’un barre-graphe à LED et piloté par USB via Simulink a été développé pour remplacer la barre originellement affichée sur l’écran de l’ordinateur. Ce dispositif est placé à côté du KEO comme le montre la figure 9.4. En plus du barre-graphe, le dispositif comporte 3 LEDs de couleur (rouge, orange et verte). La LED rouge s’allume lorsque le BS se déclenche, fournissant un retour visuel en plus du retour auditif généré par l’ordinateur. La LED orange indique que le seuil est dépassé. La LED verte sert lors de la phase d’apprentissage pour indiquer à l’utilisateur quand il doit effectuer sa tâche mentale.

Le KEO n’est pas un dispositif conçu pour être piloté par un ordinateur, mais prend en entrée un contacteur physique. Ainsi le dispositif créé par nos soins servira également d’interface avec le KEO en émulant un contacteur (grâce à un relai REED). La figure 9.5 présente la chaîne expérimentale complète.

9.2 Chaîne de Traitement

Contrairement à la première expérience où les algorithmes utilisés étaient des versions adaptatives de méthodes classiques de l’état-de-l’art, la chaîne de traitement implémentée pour cette campagne de test utilise les algorithmes basés sur la géométrie Riemannienne qui ont été développés au cours de cette thèse.

Le choix s’est porté sur un algorithme de type MDM (cf section 4.4.3) pour sa simplicité de mise en œuvre, sa rapidité de convergence et sa robustesse. L’algorithme MDM a été modifié pour s’adapter au cas particulier du BS où les deux classes (activité spécifique et non spécifique) n’ont pas la même importance. Plus particulièrement, on souhaite que les points aberrants et les artéfacts soient toujours classés comme activité non spécifique afin d’éviter les faux déclenchements de l’interrupteur. Cette variation de l’algorithme MDM, nommée Robust MDM (RMDM), inclut donc automatiquement la gestion d’artefact.

La chaîne de traitement est schématisée par la figure 9.6. Par rapport à la chaîne utilisée



FIGURE 9.4 – Le KEO (à gauche) et la carte d'interface (à droite) affichant le feedback à l'utilisateur et commandant le KEO via un relai REED.

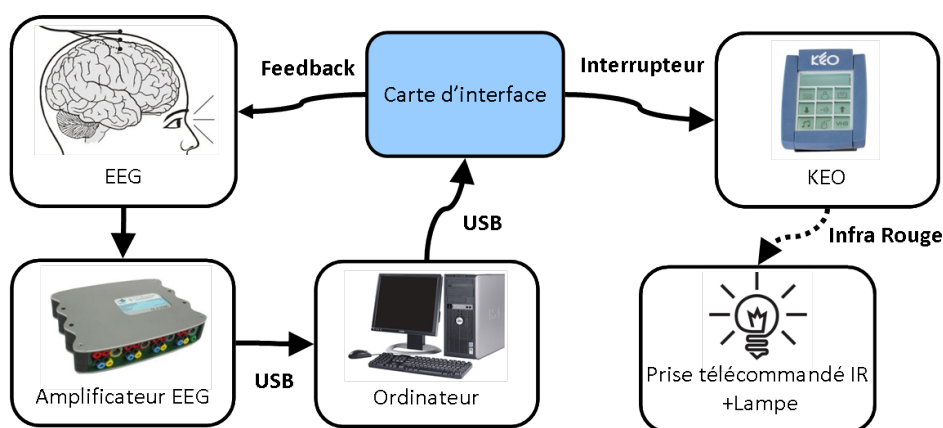


FIGURE 9.5 – Interactions entre les différents dispositifs impliqués dans la chaîne expérimentale. Les signaux EEG sont enregistrés et transmis à l'ordinateur où le traitement du signal est effectué. La sortie de ce traitement est envoyée par USB à la carte d'interface qui affiche à la fois le *feedback* et pilote le KEO. Le KEO envoie par infrarouge la commande correspondante à l'effecteur ciblé.

lors de la première campagne de test (cf figure 8.2), les blocs de détection d'artéfacts, de filtrage spatial par CSP et de classification par LDA sont remplacés par un unique algorithme. En effet l'avantage de la géométrie Riemannienne dans ce contexte est qu'elle permet de fusionner le filtrage spatial et la classification. Seul le bloc RMDM est adaptatif, ce qui simplifie grandement la mise en œuvre et accélère la convergence de la chaîne.

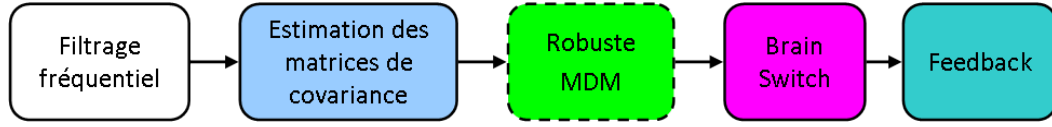


FIGURE 9.6 – Chaîne de traitement du Brain-Switch utilisant la géométrie Riemannienne. Le bloc en pointillé est adaptatif.

Filtrage fréquentiel. La fréquence d'échantillonnage est de 512 Hz et le filtrage fréquentiel consiste en un unique filtre passe bande entre 8 et 30 Hz. Il est réalisé par l'intermédiaire d'un filtre Butterworth d'ordre 5.

Estimation des matrices de covariance. Les matrices de covariance sont estimées par SCM sur une fenêtre glissante de 1 seconde, avec un décalage de 100 ms entre 2 estimations. Ainsi ce bloc fournit les matrices de covariance à une fréquence de 10 Hz. Cette fenêtre induit un délai sur l'ensemble de la chaîne de traitement et donc sur le *feedback*.

Robuste MDM. C'est le cœur de la chaîne de traitement, ce bloc effectue la classification des matrices de covariance et sa sortie sera utilisée pour déclencher ou non le BS. Nous avons développé, pour les besoins de cette expérience, un algorithme robuste à partir de l'algorithme MDM présenté section 4.4.3. Afin d'éviter les problèmes de robustesse rencontrés lors de la première campagne de test, la détection d'artefact sera effectuée au sein même de l'algorithme de classification. En utilisant un critère fondé sur la distance Riemannienne, similaire à celui employé par la classification, on s'assure ainsi que les artefacts non détectés ne pourront plus biaiser de manière significative l'apprentissage.

L'activité spécifique que l'on cherche à détecter est par définition reproductible et caractéristique. Les matrices de covariance des essais qui lui sont associées se situent donc dans une région particulière de la variété des matrices SDP. Par opposition, l'activité non spécifique sera constituée de l'ensemble des autres activités possibles, ainsi les matrices de covariance associées peuvent théoriquement compléter le reste de la variété.

la figure 9.7 montre la dispersion des matrices de covariance dans la variété (cas à deux électrodes). Les matrices correspondant à des activités spécifiques (en rouge) sont rassemblées sur une portion de la variété proche de la frontière² (symbolisée par le cône sur la figure de droite) et sont relativement peu dispersées. En revanche, les matrices correspondant aux activités non spécifiques (en bleu) sont dispersées et occupent une portion beaucoup plus large de la variété.

Comme toutes les méthodes de classification supervisée, une phase d'apprentissage permettant d'identifier les caractéristiques de l'activité spécifique est nécessaire. Dans le cas de la MDM il s'agit simplement d'estimer les matrices de covariance moyennes de chaque classe. Dans le cas de la RMDM, une région d'intérêt est définie pour se restreindre à une partie de la variété, et tout point se trouvant hors de cette région d'intérêt sera considéré comme un artefact. Le processus d'apprentissage est le suivant :

On dispose de I matrices de covariance, nommées Σ_i d'indice $i = 1 \dots I$, ainsi que de leur étiquettes associées $y_i \in \{s, \bar{s}\}$ avec s correspondant à l'activité spécifique. En premier lieu, on estime la matrice de covariance moyenne $\Sigma^{(s)}$ des matrices correspondant à l'activité spécifique par l'utilisation de la moyenne géométrique dans la variété Riemannienne définie section 4.3.3.

On définit ensuite une région d'intérêt $\Omega = \{\Sigma_i \mid \delta_R(\Sigma^{(s)}, \Sigma_i) < \epsilon\}$. La région d'intérêt est définie de manière à contenir la majeure partie des matrices correspondant à l'activité

2. Au delà de cette frontière un point de \mathbb{R}^3 ne satisfait plus les propriétés des matrices SPD, en l'occurrence la matrice ne sera plus définie positive.

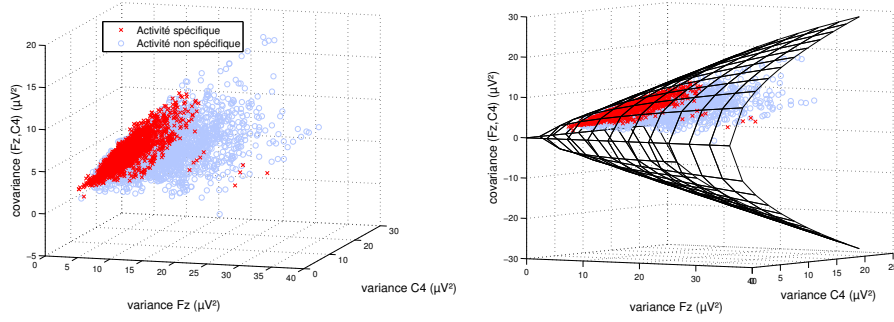


FIGURE 9.7 – Position des matrices dans la variété, cas à deux électrodes. Les bornes de la variété sont représentées par un cône sur la figure de droite.

spécifique. Ainsi, on fixe $\epsilon = m + 3s$ avec m et s respectivement la médiane et l'écart type (estimé de manière robuste par la déviation absolue à la médiane) des distances $\{\delta_R(\Sigma^{(s)}, \Sigma_i), i | y_i = s\}$, faisant ainsi l'hypothèse que ces distances suivent une distribution gaussienne. Cette approximation n'est pas rigoureuse mais permet de définir un seuil au delà duquel la probabilité d'appartenance à la classe d'activités spécifiques est pratiquement nulle.

La figure 9.8 montre un exemple de la définition de la région d'intérêt à partir de la distribution des distances. La majorité des matrices correspondant à l'activité spécifique est à l'intérieur de cette région d'intérêt, et seulement quelques-unes (probablement des artefacts) s'en trouvent exclues. On remarque également qu'un grand nombre de matrices correspondant à l'activité non spécifique sont à l'intérieur de cette région. La définition seule de la région d'intérêt ne suffit pas à discriminer les classes.

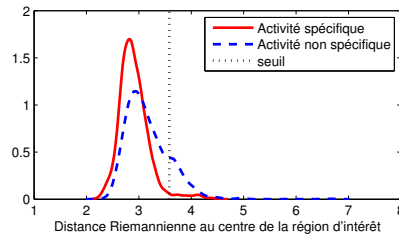


FIGURE 9.8 – Distribution des distances au centre et seuil de la région d'intérêt

Une fois cette région d'intérêt définie, on applique simplement l'algorithme MDM sur les données contenues dans la région d'intérêt. La matrice de covariance moyenne pour la classe spécifique étant déjà estimée, il suffit simplement d'estimer la matrice de covariance moyenne $\Sigma^{(\bar{s})}$ des matrices de covariance des activités non spécifiques contenues exclusivement dans la région d'intérêt.

A l'issue de cette procédure d'apprentissage, on a estimé les 3 quantités $\Sigma^{(s)}, \Sigma^{(\bar{s})}$ et ϵ qui seront utilisées pour la classification.

Toute matrice se trouvant à l'extérieur de la région d'intérêt est automatiquement considérée comme correspondant à une activité non spécifique. Si la matrice en question est à l'intérieur de la région d'intérêt, elle appartient à la classe dont la moyenne est la plus proche au sens de la distance Riemannienne. Soit Σ une matrice de classe inconnue $y \in \{s, \bar{s}\}$. Cette

prise de décision se traduit mathématiquement par :

$$y = \begin{cases} s & \text{si } \delta_R(\Sigma^{(s)}, \Sigma) < \epsilon \text{ et } \delta_R(\Sigma^{(s)}, \Sigma) < \delta_R(\Sigma^{(\bar{s})}, \Sigma) \\ \bar{s} & \text{sinon.} \end{cases} \quad (9.1)$$

Cet algorithme a été comparé à l'algorithme utilisé lors de la première campagne de test sur le jeu de données réalisé lors de cette campagne. Cette comparaison a fait l'objet d'une communication au *GRETSI 2011* [11] et à la *BCI conférence 2011* de Graz [9]. Il est ainsi montré que cet algorithme améliore la robustesse des résultats (PPV passant de 0.8 à 0.9) par rapport à une chaîne CSP + LDA, en offrant donc des résultats similaires à la chaîne CSP + LDA + détection d'artefacts employée pendant la première campagne de test.

Pour implémenter cet algorithme de manière adaptative, il suffit donc de mettre à jour les matrices de covariance moyennes et le seuil de la région d'intérêt. La procédure de mise à jour des matrices moyenne est la suivante : soit $\Sigma_{i-1}^{(s)}$ une matrice moyenne à l'itération $i-1$, et Σ_i la nouvelle matrice de covariance, la mise à jour se fait par un déplacement sur la géodésique entre $\Sigma_{i-1}^{(s)}$ et Σ_i :

$$\Sigma_i^{(s)} = (\Sigma_{i-1}^{(s)})^{\frac{1}{2}} \left[(\Sigma_{i-1}^{(s)})^{-\frac{1}{2}} \Sigma_i (\Sigma_{i-1}^{(s)})^{-\frac{1}{2}} \right]^{\frac{1}{\alpha}} (\Sigma_{i-1}^{(s)})^{\frac{1}{2}} \quad (9.2)$$

où α est un coefficient tel que :

$$\alpha = \min(i, I) \quad (9.3)$$

où la valeur I représente le nombre maximum de matrices de covariances utilisées pour le calcul et définit le pas minimum de la mise à jour. La convergence vers la moyenne géométrique de cette procédure de mise à jour a été vérifiée hors ligne sur les données de la première campagne de test. On pourra également remarquer que cette procédure de mise à jour est l'équivalent Riemannien de la mise à jour Euclidienne suivante :

$$\Sigma_i^{(s)} = \frac{\alpha - 1}{\alpha} \Sigma_{i-1}^{(s)} + \frac{1}{\alpha} \Sigma_i \quad (9.4)$$

Le seuil de la région d'intérêt est, quant à lui, mis à jour via le calcul adaptatif de la moyenne et de la variance de la distance Riemannienne au centre de la région d'intérêt. Ce calcul adaptatif est similaire à celui employé pour la détection d'artefacts dans la précédente chaîne de traitement décrite dans la Section 8.2.

Brain-switch. On utilise la sortie du classifieur pour contrôler le déclenchement du BS. Cette étape est identique à celle utilisée dans la précédente campagne décrite Section 8.

Feedback. Le retour visuel est ensuite affiché en temps réel sur le barre-graphe à LED disposé à côté du KEO. De la même manière que dans la première campagne de test, on dispose également d'un retour sonore lors du déclenchement du BS.

9.3 Campagne de Test

9.3.1 Présentation de l'expérience

Le but de cette campagne est de valider le fonctionnement d'un BS couplé avec l'effecteur KEO, et d'évaluer la capacité d'un utilisateur à prendre en main le système en un temps minimum. Il sera possible pour l'utilisateur de commander la réalisation de 3 actions :

1. Allumer une lumière (par l'intermédiaire d'une prise commandée par infrarouge via le KEO).

2. Déclencher un signal sonore (émis par le KEO).
3. commander l'allumage d'une télévision (fictive).

Pour réaliser une action, l'utilisateur devra activer une première fois le BS pour sortir le KEO de l'état de veille (ce dernier se met en veille au bout de 30 s d'inactivité), puis lancer la séquence de défilement par l'intermédiaire d'une seconde activation du BS. Après un délai de 5 s, le défilement se fait touche par touche, chaque action étant mise en surbrillance pendant 10 s. L'utilisateur dispose de 2 défilements des trois actions pour sélectionner celle qu'il souhaite avant que le KEO ne repasse au mode repos.

14 sujets ont participé à cette campagne de tests. La majorité des sujets n'avait jamais participé à une expérience BCI, excepté 2 personnes ayant participé à la précédente campagne de test qui s'était déroulée un an auparavant.

15 électrodes ont été utilisées, avec un montage resserré sur les zones motrices, la masse en AFz et la référence au niveau de l'oreille gauche. Le montage est présenté figure 9.9

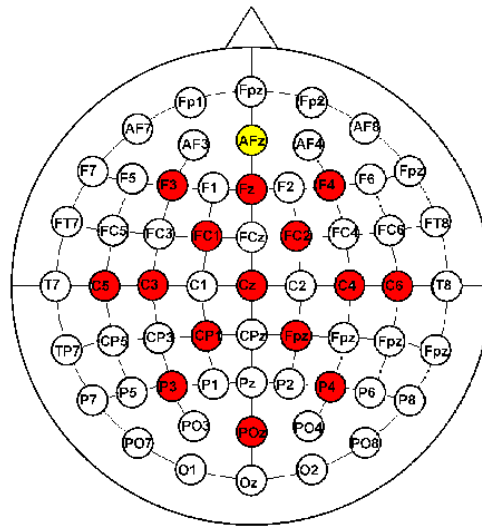


FIGURE 9.9 – Position des électrodes pour cette campagne de test. La masse est en jaune, l'électrode de référence est au niveau de l'oreille gauche.

Chaque sujet a effectué une session d'une durée maximale d'une heure (pose des électrodes comprise). Une session se déroule en 4 phases :

Prise en main. Une phase de prise en main du système initie l'utilisateur au paradigme expérimental. La tâche mentale nécessaire à l'activation du BS est l'exécution réelle d'un mouvement de la main gauche. Cette phase se termine dès que l'utilisateur s'est familiarisé avec le système. Pour les besoins de l'apprentissage, cette étape est supervisée, c'est-à-dire que le choix de l'action, que l'utilisateur doit déclencher, est dicté par l'ordinateur.

Apprentissage. Cette phase est similaire à la précédente, mais fondée sur l'imagination d'un mouvement de la main. Elle permet **d'extraire les paramètres de la chaîne de traitement qui seront utilisés par la suite**. Cette phase est de durée variable, l'expérimentateur jugeant quand il doit arrêter sur la base de la performance de l'utilisateur. Cette étape est également supervisée.

Test. Cette phase est similaire à la phase d'apprentissage, et sert à évaluer les performances de l'utilisateur à partir des paramètres fixés par l'étape d'apprentissage, elle est donc supervisée. La chaîne de traitement n'est plus adaptative (la mise à jour des paramètres est désactivée). On demande donc à l'utilisateur d'effectuer successivement 15 séquences menant à la réalisation d'une des 3 actions proposées, suivant un ordre prédéfini.

Libre. Une phase libre est proposée à l'utilisateur à la fin de la session. C'est une phase d'utilisation non supervisée où l'utilisateur choisit lui même les actions qu'il souhaite réaliser. Cette phase lui permet **d'apprécier la qualité du contrôle** qu'il notera subjectivement sur une échelle de 1 à 5 (1 : pas de contrôle sur le BS, 5 : contrôle parfait). Cette phase dure autant de temps que l'utilisateur le souhaite.

Déroulement des étapes supervisées

Parmi ces 4 étapes, 3 sont supervisées et suivent donc un déroulement particulier. Une phase supervisée se décompose en plusieurs séquences entrecoupées par des phases de repos. Une séquence correspond à une tentative de déclenchement d'une action par l'utilisateur. C'est l'expérimentateur qui initie le démarrage d'une séquence. Une séquence débute par la présentation d'une consigne, c'est-à-dire l'action que l'utilisateur doit réaliser. Cette consigne est présentée sur l'écran de l'ordinateur pendant 5 s. Suite à cette consigne, l'utilisateur dispose d'un temps raisonnable pour lancer le défilement sur le KEO. Si l'utilisateur n'arrive pas à activer le BS, l'expérimentateur peut lancer manuellement le défilement. La phase de défilement prend fin, de même que la séquence, une fois que l'utilisateur a déclenché une action, ou que le défilement est complet. Ce déroulement est schématisé par la figure 9.10.

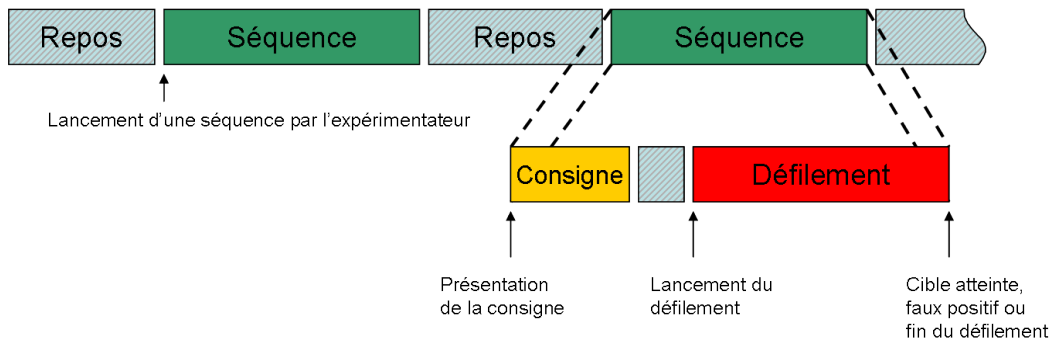


FIGURE 9.10 – Déroulement des étapes supervisées. Les séquences sont entrecoupées de phases de repos et sont lancées manuellement par l'expérimentateur. Une séquence débute par la présentation de la consigne, et se termine par le déclenchement d'une action ou la fin du défilement.

Pour ne pas perturber le déroulement de l'apprentissage, le BS ainsi que le *feedback* est activé seulement à l'issue de la consigne et jusqu'à la fin de la séquence. En effet, un faux positif en dehors de ces périodes pourrait enclencher la phase de défilement du KEO et perturber le déroulement de l'expérience. En particulier, au début de l'apprentissage quand les paramètres n'ont pas encore convergé et que le BS est très sensible. De même, la mise à jour des paramètres s'effectue uniquement pendant la phase de défilement car c'est la seule période où l'on connaît de source sûre si l'utilisateur effectue ou non la tâche mentale. Ce schéma est illustré figure 9.11.

Enfin, lors des phases d'apprentissage, la sortie du BS est également inhibée pendant les deux premières séquences ce qui permet de fournir suffisamment de données pour garantir une stabilité minimale et l'initialisation des paramètres.

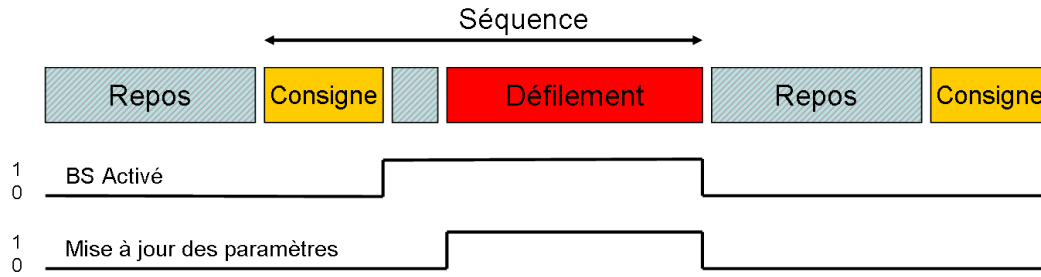


FIGURE 9.11 – L’activation du BS se fait uniquement après la consigne, et la mise à jour des paramètres uniquement pendant la phase de défilement.

9.3.2 Évaluation des performances :

Pour cette campagne de test, l’accent a été mis sur la vitesse d’apprentissage et sur les performances de l’application.

Phase d’apprentissage. Le temps nécessaire à l’apprentissage sera évalué à partir des jeux de données d’apprentissage. On pourra également observer les caractéristiques spatiales extraites à partir des covariances moyennes des deux classes (activité spécifique vs. activité non-spécifique)

Phase de test. Les performances de l’application seront évaluées sur les données de test. Dans cette campagne, le déclenchement du BS est le moyen de réaliser une action, et non le but de l’expérience comme c’est le cas pour la première campagne décrite chapitre 8. Les performances ne seront donc pas évaluées sur la base du BS mais plutôt sur la réalisations des actions. On s’intéressera également sur le temps nécessaire à la réalisation de chacune des 3 actions. Enfin, on étudiera plus en détail les déclenchements asynchrones (sortie de veille et lancement du défilement).

La phase de test comporte 15 séquences où l’on demande à l’utilisateur de réaliser une action particulière parmi la liste des 3 actions disponibles. Si l’utilisateur échoue à réaliser l’action demandée, par exemple à cause d’un faux déclenchement du BS, la séquence suivante lui demandera de réaliser la même action. Cela correspond à un cas de figure réel où l’utilisateur tente de réaliser la même action tant que cette tentative n’est pas couronnée de succès. Ainsi les utilisateurs n’ont pas tous effectué le même enchaînement de séquences. On s’intéressera donc ici au nombre d’actions correctement effectuées, sur les 15 actions. On observera également le nombre médian de tentatives nécessaire à la réalisation de l’action voulue.

Phase libre. Enfin, à la fin de l’expérience, on demande à l’utilisateur de noter la qualité du contrôle qu’il a pu expérimenter pendant la phase libre. Cette note fera office de mesure qualitative de la qualité du contrôle ressenti par l’utilisateur.

9.3.3 Résultats

Parmi les 14 sujets ayant participé à la campagne, 2 n’ont pas obtenu de résultats significatifs à la fin de la phase d’apprentissage, et n’ont donc pas poursuivi l’expérience. Il s’agit probablement du phénomène d’illettrisme ICM (*BCI illiteracy*) décrit dans la littérature et qui touche 15 à 30% de la population [29]. Nous sommes ici à un pourcentage de 15%. Ces deux sujets sont donc retirés de l’étude ce qui porte le nombre de participants à 12.

Phase d'apprentissage. Le tableau 9.1 présente les temps d'apprentissage, noté TA, ainsi que le nombre de séquences, noté #S, réalisées pour chaque utilisateur.

Utilisateur	TA (minutes)	#S
AB	9.1	10
AC	10.3	10
AD	8.9	10
AE	13.2	15
AF	9.3	13
AH	6.2	9
AI	10.9	9
AJ	4.5	5
AK	5.5	8
AL	7.0	10
AN	10.1	10
AO	7.6	9
Moyenne	8.5	9.8

TABLE 9.1 – Temps de la phase d'apprentissage (TA) et nombre de séquences (#S) pour les 12 utilisateurs.

On observe ainsi un TA moyen de 8 min et 30 s, avec un minimum de 4 min et 30 s pour l'utilisateur AJ et un maximum de plus de 13 min pour l'utilisateur AE. Le nombre moyen de séquences est de 10. Ce temps est très court compte tenu du faible nombre de tentatives de déclenchement réalisé par l'utilisateur. En effet, on compte environ 1 séquence par min ce qui équivaut à 2 ou 3 tentatives de déclenchement du BS par minute. L'expérience comporte un grand nombre de phases de repos, en comparaison de la précédente campagne où le nombre de tentatives de déclenchement oscillait entre 3 et 6 déclenchements par minute.

La figure 9.12 montre la caractéristique spatiale dominante pour les utilisateurs AC et AJ. On note clairement l'activation de la zone motrice dédiée à la main droite pour ces deux utilisateurs, ce qui est en accord avec la tâche mentale réalisée.

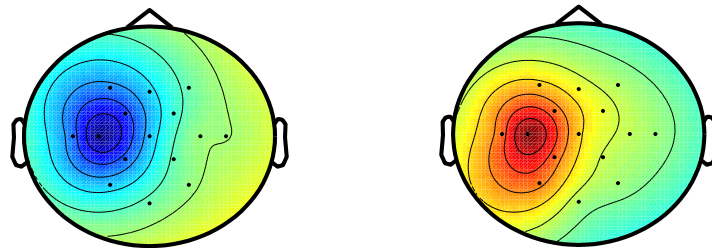


FIGURE 9.12 – Caractéristiques spatiales principales pour l'utilisateur AC (à gauche) et AJ (à droite) à la sortie de la phase d'apprentissage.

Enfin, par rapport à la première campagne de test où l'apprentissage de plusieurs utilisateurs a été influencé par la présence d'artefacts non détectés, ce type de problème ne s'est pas reproduit. La chaîne de traitement est donc plus robuste.

Phase de test. La phase de test permet de mesurer les performances de l'application. La figure 9.13 illustre la réalisation d'une séquence pour l'utilisateur AJ. La séquence débute au

temps $t = 450s$, l'utilisateur patiente ensuite quelques secondes et déclenche une première fois le BS ($t = 467s$) pour sortir le KEO de l'état de veille. L'utilisateur déclenche ensuite une deuxième fois le BS pour lancer le mode de défilement qui lui présentera successivement les différentes actions. Lorsque l'action ciblée est mise en surbrillance par le défilement ($t = 476s$), l'utilisateur tente de déclencher le BS pour valider cette action. Ce dernier déclenchement marque la fin de la séquence.

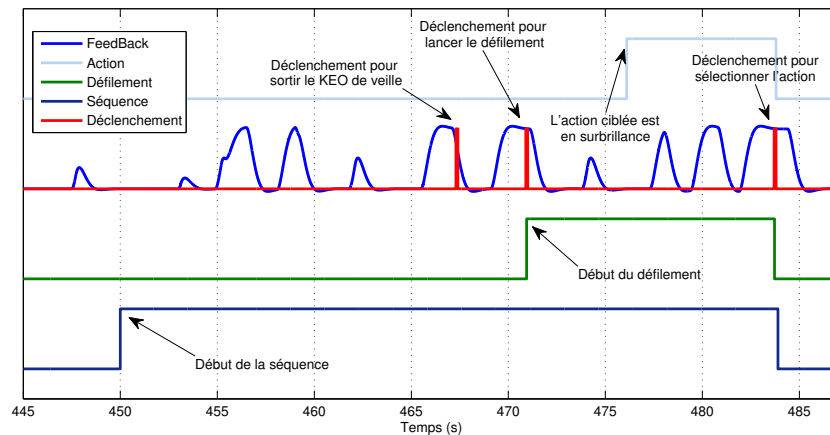


FIGURE 9.13 – Évolution temporelle d'une séquence pour le sujet AJ. On note les 3 déclenchements du BS permettant la sélection et la réalisation d'une action.

La figure 9.14 permet d'observer l'efficacité de la détection d'artéfacts. Sur la période de temps observée (80 s), on remarque que la distance au centre de la région d'intérêt augmente brusquement à plusieurs reprises et dépasse le seuil fixé lors de l'apprentissage. L'observation des signaux EEG sur un sous-ensemble d'électrodes permet de confirmer visuellement la présence d'artéfacts lors de ces brusques augmentations.

La Table 9.2 présente les résultats des 15 séquences pour les différents utilisateurs. Le caractère ✓ signifie que l'utilisateur a réussi à réaliser l'action qui lui était demandée dans la consigne. Le caractère ✗ signifie que l'action réalisée est la mauvaise, ou qu'aucune action n'a été réalisée.

Certain utilisateurs s'en sortent très bien, avec 14/15 des actions correctement réalisées (utilisateur AO et AJ), à la différence des utilisateurs AD et AL.

Le cas de l'utilisateur AL est très intéressant. En observant le déroulement de la phase de test, donnée figure 9.15, on peut faire plusieurs constatations :

- Le *feedback* est très "propre", dans le sens où il oscille très peu. En particulier pendant les phases de repos entre deux séquences, le *feedback* reste à zéro. Ceci signifie que l'algorithme de classification arrive parfaitement à classifier l'activité non spécifique.
- Tous les déclenchements asynchrones du BS sont réalisés avec succès. De même lorsque l'action ciblée est mise en surbrillance, l'utilisateur arrive toujours à déclencher le BS. L'utilisateur parvient donc correctement à contrôler le déclenchement du BS quand il le souhaite.
- Les faux déclenchements de l'interrupteur interviennent uniquement lors des phases de défilement.

On peut donc conclure à partir de ces constatations que la phase de défilement perturbe l'utilisateur et produit des faux déclenchements du BS. Le changement de contexte entre les phases asynchrones et les phases synchrones modifie généralement l'activité mentale du sujet qui éprouve ainsi plus de difficulté à contrôler le BS.

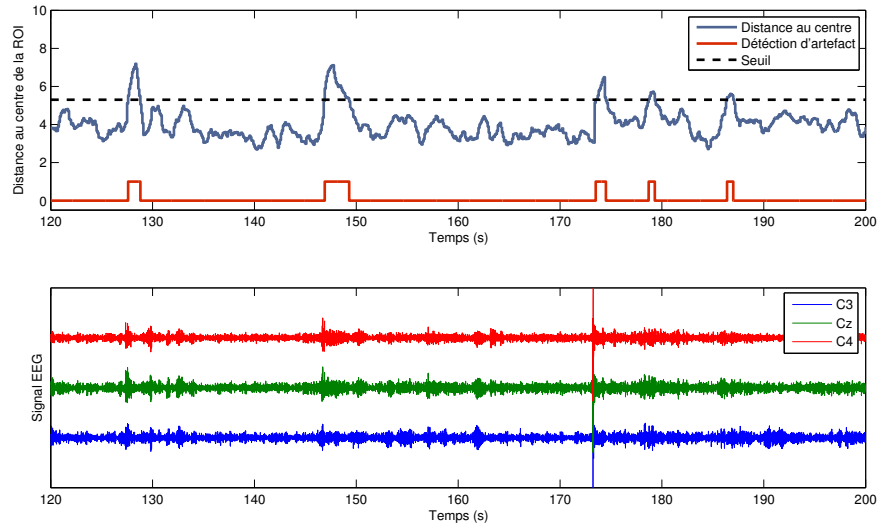


FIGURE 9.14 – Détection d’artéfacts (en haut) et signal EEG sur les électrodes C3, Cz et C4 (en bas). La présence d’artéfact sur le signal est bien corrélée avec la détection et la distance au centre de la région d’intérêt.

Si l’on observe la caractéristique spatiale principale obtenue lors de l’apprentissage, donnée figure 9.16, on remarque que les zones impliquées n’ont rien à voir avec la tâche mentale (mouvement imaginé de la main droite). Ainsi, c’est préférentiellement les zones frontales qui sont impliquées. L’explication la plus probable est donc que l’apprentissage a probablement dû accrocher sur l’état mental de concentration lié à la réalisation de la tâche mentale plutôt que sur cette dernière. Ainsi l’utilisateur arrive simplement à déclencher le BS en se concentrant. Cela fonctionne très bien pour les déclenchements asynchrones, en revanche lorsque l’on passe en mode défilement, qui implique une plus grande charge mentale, cela aboutit à des déclenchements involontaires du BS.

Les performances globales sont données Table 9.3, en terme de nombre d’actions correctement réalisées, notées **#AC** et **%AC** pour le pourcentage. Cette table donne également le temps nécessaire à la réalisation des 15 séquences, ainsi que le nombre médian de tentatives nécessaire à la réalisation d’une action, noté **Med NT**. Enfin, elle fournit la valeur du TPR pour les déclenchements de type asynchrones.

On obtient ainsi un score moyen de 68 % d’actions correctement réalisées, c’est-à-dire de séquences ayant conduit à la réalisation de l’action demandée par la consigne. On remarque également 4 utilisateurs (AF, AJ, AN et AO) dont les scores sont particulièrement bons avec environ 90% de réussite.

On note en revanche 3 utilisateurs (AC, AD et AL) dont les performances ($\leq 50\%$) ne sont pas suffisantes pour envisager une utilisation dans le cadre de la vie courante.

Le TPR observé pour les déclenchements asynchrones est en moyenne de 0.9. Le BS offre donc des performances similaires à celles observées lors de la première campagne de test (cf chapitre 8). On remarquera que cette valeur est généralement élevée même pour les utilisateurs n’ayant pas réussi à réaliser toutes les actions. Il est donc plus facile pour l’utilisateur de sortir de l’état de veille (ou de lancer le défilement) que de sélectionner l’action désirée au bon moment. On peut en conclure que le timing du défilement n’était pas optimal pour tous les sujets.

Il est intéressant d’observer le nombre médian de tentatives nécessaire à la réalisation

User	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
AB	✗	✓	✓	✓	✓	✗	✗	✗	✓	✓	✓	✗	✓	✓	✗
AC	✗	✓	✓	✗	✗	✗	✗	✓	✓	✗	✓	✓	✗	✓	✓
AD	✗	✗	✓	✓	✗	✓	✗	✗	✗	✓	✗	✗	✓	✓	✗
AE	✗	✓	✓	✓	✓	✗	✗	✓	✓	✓	✓	✗	✗	✓	✗
AF	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓
AH	✓	✓	✗	✓	✗	✓	✗	✓	✓	✗	✗	✗	✓	✗	✓
AI	✓	✓	✓	✓	✓	✓	✗	✗	✓	✓	✓	✓	✗	✗	✓
AJ	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓
AK	✓	✓	✓	✓	✗	✗	✓	✗	✓	✓	✓	✓	✗	✓	✓
AL	✓	✓	✗	✓	✗	✓	✗	✗	✗	✗	✓	✗	✓	✗	✗
AN	✗	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
AO	✓	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓

TABLE 9.2 – Succès ✓ ou échec ✗ dans la réalisation de l'action pour chacune des 15 séquences de la phase de test.

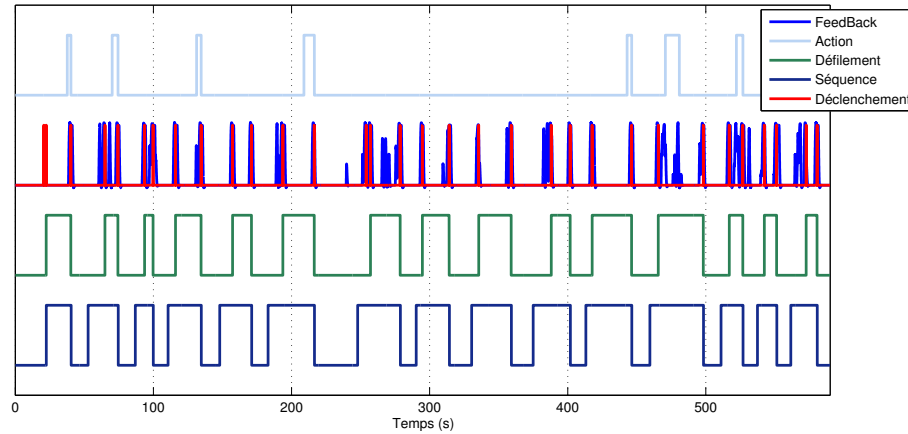


FIGURE 9.15 – Évolution temporelle de la phase de test pour l'utilisateur AL

d'une action, noté **Med NT**. C'est un paramètre important pour l'utilisateur. Par exemple, on observe sur le tableau 9.2 que l'utilisateur AB a eu besoin de 4 séquences pour réaliser l'action demandée lors de la séquence 6. L'action placée en 3ème position dans le défilement est plus dure à réaliser car la probabilité d'avoir un faux déclenchement pendant le défilement des autres actions est plus grande. Il est parfois nécessaire de s'y reprendre à plusieurs fois avant d'atteindre cette action. Ainsi, 8 des 12 utilisateurs ont un nombre médian de tentatives de 1, c'est-à-dire qu'à quelques exceptions près, il leur a suffi d'une seule tentative pour réaliser l'action désirée.

En moyenne, le nombre de tentatives est de 1.3, ce qui montre que malgré le taux de réussite de 68%, les utilisateurs arrivent relativement rapidement à réaliser l'action désirée.

La figure 9.17 présente la distribution du temps nécessaire au premier déclenchement du BS dans le mode asynchrone, c'est-à-dire pour sortir le KEO de l'état de veille ou de lancer le défilement. Cette quantité représente le temps qu'a mis l'utilisateur pour déclencher le BS après que la consigne lui ait été présentée. Le maximum de cette distribution se trouve autour de 4 s, ce qui est plutôt rapide compte tenu du délai de 1 s induit par l'estimation des matrices de covariance ainsi que celui de 1 s nécessaire pour activer le BS. On note également un rebond autour des 12 s. En effet, il est courant que l'utilisateur n'arrive pas à

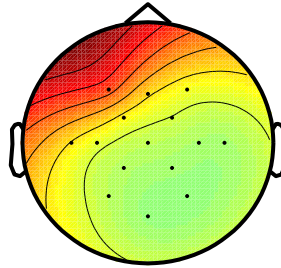


FIGURE 9.16 – Caractéristique spatiale principale pour l'utilisateur AL.

User	TPR	#AC	%AC	Temps (min)	Med NT
AB	1	9	60	12.0	1
AC	0.71	8	53	8.9	1.5
AD	0.93	6	40	10.8	2.5
AE	0.79	9	60	10.4	1
AF	0.71	13	87	9.6	1
AH	1	8	53	11.2	2
AI	0.86	11	73	11.9	1
AJ	1	14	93	11.0	1
AK	1	11	73	8.6	1
AL	1	6	40	9.8	2
AN	0.87	13	87	12.9	1
AO	1	14	93	11.2	1
Moyenne	0.9	10.2	68	10.7	1.3

TABLE 9.3 – Résultats chiffrés pour les 12 sujets. TPR pour les déclenchement asynchrones uniquement, #AC le nombre d'actions correctement réalisées, %AC le pourcentage correspondant et Med NT le nombre médian de tentative de déclenchement du BS.

déclencher le BS lors de sa première tentative de réalisation de sa tâche mentale. Il marque donc généralement une courte pause puis recommence sa tâche mentale.

La figure 9.17 présente également la fonction de répartition de ce temps nécessaire au premier déclenchement. On note ainsi que 80% des déclenchements se produisent en moins de 10 s.

Phase libre. La phase libre (sans consigne) permet à l'utilisateur d'apprécier le contrôle de l'application en choisissant lui-même les actions qu'il souhaite réaliser. A l'issue de cette phase, l'utilisateur attribue une note entre 1 et 5 à la qualité du contrôle qu'il a ressenti. La table 9.4 présente ces résultats.

User	AB	AC	AD	AE	AF	AH	AI	AJ	AK	AL	AN	AO	Moyenne
Note	4	4	3	3	4	3.5	4	4	3.5	3	4	4	3.7

TABLE 9.4 – Note attribuée par chaque utilisateur à la qualité du contrôle de l'application lors de la phase libre.

On obtient une note moyenne de 3.7, ce qui dénote un contrôle plutôt bon de l'appli-

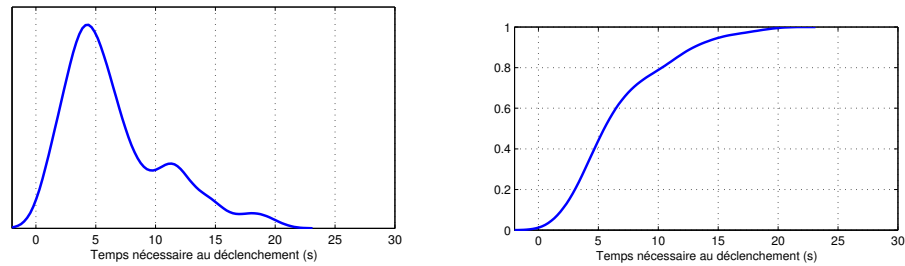


FIGURE 9.17 – Distribution (à gauche) et fonction de répartition (à droite) du temps nécessaire au premier déclenchement du BS dans le mode asynchrone, c'est-à-dire pour sortir le KEO de veille ou lancer le défilement.

cation. Les utilisateurs ayant donné la note de 3 font part de la remarque suivante : le *feedback* oscille de manière assez importante, ils n'ont donc pas l'impression de contrôler particulièrement l'évolution du *feedback*. En revanche, ils contrôlent bien le déclenchement du BS en profitant de ces oscillations et en réalisant la tâche mentale lorsque le *feedback* est déjà au dessus du seuil, lui permettant de se maintenir le temps requis pour le déclenchement. Les utilisateurs ayant noté 4 ont un contrôle plus propre et direct du BS, malgré quelques faux déclenchements, ils ont réellement le contrôle sur le *feedback* et donc sur l'application.

9.4 Conclusion

Cette deuxième campagne de test, en poursuivant le travail réalisé lors de la première campagne, a permis de mettre en œuvre avec succès un couplage du BS avec un effecteur, en offrant la possibilité à ses utilisateurs d'interagir avec l'environnement immédiat (allumer une lumière, etc ...).

L'utilisation d'un algorithme moins sensible aux artéfacts a permis de résoudre les principaux problèmes de stabilité de l'apprentissage adaptatif rencontrés lors de la première campagne de test.

Les performances globales sont satisfaisantes (68% d'action correctement réalisées), avec un certain nombre d'utilisateur offrant même de très bons résultats (> 90%), avec seulement un apprentissage de 10 minutes.

Quatrième partie

Conclusion

Conclusion et perspectives

Dans ce chapitre, nous concluons en revenant sur les principaux travaux réalisés au cours de ces 3 ans. Les avantages et les inconvénients des différentes approches seront discutés afin de proposer les pistes de recherches pour les développements futurs.

Conclusions

Au cours de cette thèse et de ce manuscrit, nous avons tenté d'aborder de manière cohérente l'ensemble des points nécessaires à la réalisation d'une ICM, en gardant à l'esprit les contraintes liées à une mise en œuvre dans un contexte de vie courante. Le travail réalisé lors de cette thèse s'articule en deux parties. La première partie concerne l'**algorithmie**, le but a été de développer des méthodes de traitement du signal originales afin de résoudre les problématiques liées à l'utilisation des ICMs dans un contexte de vie courante. La deuxième partie concerne la **réalisation d'expérimentations** et la mise en œuvre d'une ICM utilisant les méthodes développées jusqu'à présent. Ces deux tâches ont été conduites en parallèle. La possibilité de réaliser des expériences **conjointement** avec le développement algorithmique est un point clef dans la réalisation d'une ICM. En effet, elle permet une interaction plus grande entre les retours de la part des utilisateurs et la conception de la chaîne de traitement orientant ainsi le développement algorithmique vers des problématiques directement liées à l'expérimentation.

Dès le début de la thèse, nous nous sommes concentrés sur l'exploitation de l'information spatiale contenue dans les signaux EEG. Bien que cette problématique soit très largement traitée dans la littérature, nos premières expérimentations nous ont montré que de nombreuses améliorations pouvaient encore être apportées, notamment sur le volet de la robustesse. Les principales techniques de **filtrage spatial** ont donc été étudiées en profondeur et mises en œuvre expérimentalement. Cette étude a fait ressortir le rôle primordial que joue la **matrice de covariance** dans l'exploitation de cette information spatiale. L'utilisation de la **géométrie de l'information**, qui permet de manipuler ces matrices en respectant la topologie de leur espace, est alors apparue comme évidente pour traiter la problématique de classification des signaux EEG. Le concept de la géométrie de l'information s'imisce à tous les niveaux de la chaîne de traitement, que ce soit pour la classification [8, 10], pour la sélection d'électrodes [6] ou de filtres spatiaux [7], pour apporter de la robustesse lors des expérimentations [9, 11] ou finalement pour aborder avec une nouvelle vision la chaîne de traitement de référence [7].

Classification

Les méthodes de classification développées reposent sur l'emploi d'une **nouvelle métrique** permettant de comparer entre elles les matrices de covariance. Cette métrique **Riemannienne** est en réalité la métrique de la variété des distributions normales multivariées et centrées définie dans le cadre de la géométrie de l'information. Un panel de méthodes, de complexité variée, a donc été proposé et se divise en 3 groupes :

Les méthodes de classification dans la variété (MDM et kNN) offrent des performances identiques aux méthodes de l'état de l'art tout en étant très simples à mettre en œuvre.

Les méthodes de filtrage géodésique permettent de dé-bruiter les données avant de leur appliquer une classification dans la variété. On pourra les voir comme des méthodes de filtrage spatial non-linéaire. Les performances obtenues surpassent nettement les méthodes

de l'état de l'art sur l'ensemble des jeux de données testés. Elles représentent un bon compromis entre complexité et performances.

Les méthodes de classification dans le plan tangent permettent d'utiliser des algorithmes de classification évolués (Régression logistique, SVM) pour un gain en performance significatif. Ce sont les méthodes les plus prometteuses, le passage dans le plan tangent permettant d'appliquer aux matrices de covariance l'ensemble des méthodes de classification couramment utilisées dans les espaces Euclidiens. En revanche, ces méthodes se confrontent au problème de classification en grande dimension, qui devient limitant lorsque le nombre d'électrodes augmente.

Les méthodes des deux premiers groupes peuvent être employées **sans modification** dans la majorité des cas, y compris sur des données fortement bruitées. Elles sont également très simples à implémenter. Elles peuvent être directement utilisées par des non-spécialistes car elles ne nécessitent le réglage d'aucun paramètre et sont très stables. Lors de leur application sur l'ensemble des données que nous avons eu à disposition, nous n'avons pas observé de désavantage à leur emploi, si ce n'est un coût calculatoire plus élevé.

Les méthodes de classification dans l'espace tangent nécessitent encore des améliorations avant de pouvoir prétendre à une large acceptation de la part de la communauté des ICMs. D'une part, elles requièrent l'optimisation d'un certain nombre de paramètres, ce qui demande généralement la maîtrise de la méthode en question. D'autre part, elles sont plus sensibles à l'augmentation du nombre d'électrodes et doivent donc être employées avec précaution.

Sélection d'électrodes

La problématique de sélection d'électrodes a également été abordée et deux algorithmes ont été développés. L'un d'eux se fonde directement sur les travaux concernant la géométrie Riemannienne et offre des performances remarquables. L'emploi d'algorithmes de sélection d'électrodes reste néanmoins anecdotique dans les expérimentations menées jusqu'à présent dans le domaine. Cependant, ce constat devrait être amené à changer dans le futur, car l'utilisation de tels algorithmes, notamment dans le cadre de la sélection spécifique à une application, répond à une problématique réelle pour qui souhaite réaliser une ICM à destination du grand public.

Expérimentations

Lors de cette thèse, nous avons conduit deux campagnes de tests portant sur la mise en œuvre d'un *brain-switch*, dont la réalisation a été guidée par les critères de **robustesse** et **d'ergonomie**. Sur ces deux points, les résultats obtenus sont encourageants. La première campagne avait pour but de valider l'utilisation d'un paradigme *asynchrone* à apprentissage adaptatif. Il a été montré que l'utilisateur pouvait ainsi apprendre à contrôler un *brain-switch* avec un apprentissage de 10 minutes environ avec un bon taux de réussite, équivalent à celui observé dans l'état de l'art. L'originalité de cette expérience réside aussi dans la possibilité laissée à l'utilisateur de choisir et d'adapter au mieux et en fonction du feedback la tâche mentale qu'il va utiliser pour contrôler le *brain-switch*. Cette liberté, rendue possible par l'emploi d'une chaîne adaptative et d'un *feedback* à faible latence (250 ms), stimule le processus d'apprentissage de l'utilisateur et le replace au centre de l'expérience. Ainsi, certains sujets ont modifié la tâche mentale initiale de manière très significative, au point d'éprouver des difficultés à décrire la tâche mentale utilisée en fin d'expérience. En revanche, les algorithmes de l'état de l'art (CSP et LDA) se sont montrés trop sensibles aux artefacts, ce qui a perturbé le déroulement de l'expérience pour plusieurs des sujets.

La deuxième campagne de test a permis de coupler ce *brain-switch* à un effecteur par l'intermédiaire d'un dispositif de contrôle d'environnement, le KEO, couramment utilisé par

des personnes en situation d'handicap. Ce dispositif se pilote simplement par l'intermédiaire d'un interrupteur et offre différents modes de navigation afin de permettre à l'utilisateur de choisir une action à réaliser dans un menu. Nous avons ainsi offert la possibilité de réaliser 3 actions : allumer une lampe, émettre un signal sonore et allumer une télévision. Avec un apprentissage de 5 à 15 minutes, la majorité des utilisateurs a ainsi pu interagir avec succès avec leur environnement. L'utilisation d'un algorithme robuste fondé sur la géométrie Riemannienne a permis de simplifier grandement la chaîne de traitement et l'a rendue moins sensible aux artefacts, les problèmes rencontrés lors de la première campagne ne se sont pas reproduits.

Paradoxalement, c'est le couplage avec l'effecteur et non la mise en œuvre de l'algorithme de classification qui a posé le plus de problèmes. En effet, le KEO n'est pas conçu pour être piloté par un ordinateur, et pour les besoins de l'apprentissage supervisé, il est nécessaire de connaître l'état dans lequel se trouve le dispositif à chaque instant. Ainsi un KEO "virtuel" (un observateur) a été introduit, reproduisant le comportement du KEO et permettant ainsi d'accéder aux variables internes (action sélectionnée, veilles, etc).

Cependant, un certain nombre de points n'ont pu être abordé au cours de ces expérimentations. La principale interrogation concerne la stabilité de la classification au cours du temps (pour des sessions plus longues) ainsi que sa réutilisation entre différentes sessions. Malgré un temps d'apprentissage très court, il n'est pas réellement envisageable, pour une utilisation quotidienne, de devoir effectuer une procédure d'apprentissage complète à chaque mise en place de dispositif.

Perspectives

Calibration et adaptation. De manière générale, l'abandon du filtrage spatial est un grand pas vers la simplification de la chaîne de traitement des ICMs. L'exploitation directe de l'information spatiale par la méthode de classification, sans passer par le filtrage spatial, ouvre de nombreuses perspectives. Ainsi, la phase de calibration de l'ICM pourrait encore être réduite en initialisant la classification avec un classifieur commun à l'ensemble des utilisateurs, qui serait ensuite adapté spécifiquement à chaque utilisateur. L'estimation de ce classifieur est alors facilitée par la simplification de la chaîne de traitement. L'adaptation inter-utilisateurs ou inter-sessions des paramètres de la classification pourrait être traitée par l'intermédiaire de méthodes non supervisées, les résultats préliminaires sur l'algorithme de classification non-supervisée k-means montrant de bonnes performances (dans certains cas). De la même manière, cette adaptation est facilitée par la propriété d'invariance par projection de la métrique Riemannienne. En effet, les relations de distances dans la variété sont conservées par toute projection linéaire de $Gl(n)$. Sous l'hypothèse que les variabilités inter-sessions soient modélisables par une telle application linéaire, l'adaptation reviendrait, dans le cas de la classification dans l'espace tangent, à une simple mise à jour du point de calcul de l'espace tangent. Des résultats préliminaires encourageants ont été obtenus par cette méthode.

Extension des méthodes Riemannienne à d'autres paradigmes. Dans cette thèse, la géométrie Riemannienne est utilisée uniquement dans le cadre de paradigmes fondés sur l'imagerie motrice, mais ce n'est pas une limitation intrinsèque. Ainsi, son emploi peut être étendu à d'autres applications de l'EEG, comme la détection des phases de sommeil ou de l'estimation de la charge mentale. D'une manière générale, ces méthodes sont applicables partout où les signaux sont caractérisés par leur variance. En revanche, elles sont peu adaptées à la détection des potentiels évoqués du type P300, l'estimation de la matrice de covariance spatiale ne prenant pas en compte la dépendance des échantillons temporels successifs. Il est néanmoins envisageable de les appliquer sur de tels paradigmes, via une

adaptation de l'estimation des matrices de covariance dans le but de faire ressortir les caractéristiques temporelles des signaux.

Exploitation de l'information fréquentielle. Ces travaux sur l'exploitation de l'information spatiale se sont fait au détriment de l'exploitation de l'information fréquentielle. Si l'on souhaite augmenter les performances de manière significative, il est nécessaire d'exploiter des informations complémentaires, notamment celles contenues dans le domaine fréquentiel. Ainsi, on peut espérer un gain important en utilisant des matrices de covariances spatio-fréquentielle afin d'exploiter les évolutions conjointes de variance entre les différentes électrodes et bandes de fréquences. Nos recherches préliminaires effectuées dans ce sens (utilisation de banque de filtres, décomposition de matrices en produit de Kronecker, ...) ont été peu fructueuses car l'augmentation de la dimension de la variété pénalisait généralement les résultats obtenus. Néanmoins, c'est dans ce sens que les futurs travaux devront être menés, afin de supprimer, après le filtrage spatial, le filtrage fréquentiel.

Cinquième partie

Annexes

Estimation des matrices de Covariance

A.1 Introduction

La matrice de covariance tient un rôle central dans la plupart des méthodes présentées tout au long de ce document. Son estimation correcte est donc absolument indispensable.

Soit $\mathbf{x}(t) \in \mathbb{R}^C$ un échantillon temporel de signal EEG enregistré sur C électrodes. On suppose que $\mathbf{x}(t)$ est la réalisation d'un processus aléatoire suivant une loi normale multivariée $\mathcal{N}(\mu, \Sigma)$. La matrice de covariance est égale à :

$$\Sigma = \mathbb{E} \left[(\mathbf{x}(t) - \mathbb{E}[\mathbf{x}(t)]) (\mathbf{x}(t) - \mathbb{E}[\mathbf{x}(t)])^T \right] \quad (\text{A.1})$$

Or on ne dispose que de N_t réalisations de ce processus aléatoire pour estimer au mieux cette matrice de covariance. Soit $\mathbf{X} \in \mathbb{R}^{C \times N_t}$ la matrice constituée de ces N_t réalisations (des échantillons temporels dans le cas de l'EEG), on cherche une fonction d'estimation $f(\cdot)$ tel que :

$$\hat{\Sigma} = f(\mathbf{X}) \simeq \Sigma \quad (\text{A.2})$$

avec $\hat{\Sigma}$ une **estimation** de la valeur vraie de la covariance.

Plusieurs quantités permettent de quantifier la qualité d'un estimateur. Étant donné que $\hat{\Sigma}$ est obtenue à partir d'un tirage de N_t réalisations, $\hat{\Sigma}$ est elle-même une réalisation d'un processus aléatoire. Alors il est possible de définir des notions statistiques sur les estimateurs, dont deux sont particulièrement importantes :

Le Biais : Le biais d'un estimateur est égal à l'écart entre la valeur vraie et l'espérance de l'estimateur :

$$B(\hat{\Sigma}) = \Sigma - \mathbb{E}[\hat{\Sigma}] \quad (\text{A.3})$$

On parle d'estimateur non-biaisé lorsque le biais est nul. Un estimateur non-biaisé a plus de chance de s'approcher de sa valeur vraie et donc donne en moyenne une estimation de meilleure qualité qu'un estimateur biaisé. En effet l'estimateur non-biaisé minimisera l'erreur quadratique $\mathbb{E}[(\hat{\Sigma} - \Sigma)^2]$ entre l'estimation et la valeur vraie.

La Convergence : Un estimateur est dit convergent (ou parfois consistant) s'il converge vers sa valeur vraie quand le nombre de réalisations utilisées dans l'estimation tend vers l'infini.

$$\hat{\Sigma} \xrightarrow[N_t \rightarrow \infty]{} \Sigma \quad (\text{A.4})$$

Robustesse : Un autre paramètre important des estimateurs est leur **robustesse**. La robustesse peut se définir comme étant la résistance d'un estimateur aux points aberrants (outliers), ou en d'autres termes la capacité de l'estimateur à estimer la bonne valeur malgré la présence de réalisations $\mathbf{x}(t)$ suivant une loi dont les paramètres divergent fortement de ceux qu'on souhaite estimer.

On utilise généralement le **point de rupture** comme mesure de la robustesse. Le point de rupture représente le pourcentage de contamination maximum au delà duquel l'estimateur donne un résultat aberrant. Ainsi un estimateur aura un point de rupture de 5% s'il n'est plus capable d'estimer Σ avec précision lorsque \mathbf{X} est contaminé par plus de 5% de points appartenant à une distribution dont la covariance est significativement différente de Σ .

Le point de rupture maximum est de 50% et il existe des estimateurs qui s'en rapprochent.

La figure A.1 montre un mélange de deux distributions $\mathcal{N}_1 = \mathcal{N}(0, 1)$ et $\mathcal{N}_2 = \mathcal{N}(6, 1)$ à hauteur de 10%. On souhaite estimer la moyenne $\mu = 0$ de \mathcal{N}_1 , les données étant alors contaminées par 10% de points appartenant à \mathcal{N}_2 , l'utilisation de la moyenne empirique est fortement déconseillée car elle dévie fortement comme on peut le voir. La médiane est alors un estimateur plus robuste de μ .

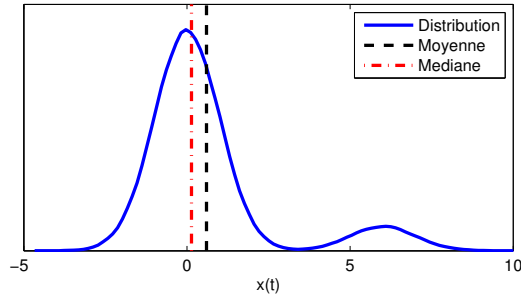


FIGURE A.1 – mélange de deux distributions $\mathcal{N}_1 = \mathcal{N}(0, 1)$ et $\mathcal{N}_2 = \mathcal{N}(6, 1)$ à hauteur de 10%. La médiane est un estimateur de μ plus robuste que la moyenne empirique.

A.2 Estimation

Il existe de nombreux estimateurs pour la matrice de covariance. Nous dresserons ici une liste non exhaustive des principaux. On considèrera dans tout ce qui suit que les données sont centrées (de moyenne nulle), comme c'est le cas en EEG.

Sample Covariance Matrix (SCM)

C'est l'estimateur classique utilisé dans la plupart des algorithmes. Il est non-biaisé, convergent et très peu robuste. L'équation est la suivante :

$$\hat{\Sigma} = \frac{1}{N_t - 1} \mathbf{X} \mathbf{X}^T \quad (\text{A.5})$$

Normalized Sample Covariance Matrix (NSCM)

Cette estimateur est une variation autour de la SCM. Il est biaisé et non convergent. En revanche il est plus robuste car il pénalise les points dont les valeurs sont grandes.

$$\hat{\Sigma} = \frac{C}{N_t} \sum_{t=1}^{N_t} \frac{\mathbf{x}(t) \mathbf{x}(t)^T}{\mathbf{x}(t)^T \mathbf{x}(t)} \quad (\text{A.6})$$

Fixed Point estimator (FP)

Cet estimateur est similaire à l'estimateur NSCM, mais la normalisation est effectuée par l'intermédiaire de la distance de Mahalanobis :

$$\hat{\Sigma} = \frac{C}{N_t} \sum_{t=1}^T \frac{\mathbf{x}(t)\mathbf{x}(t)^T}{\mathbf{x}(t)^T \hat{\Sigma}^{-1} \mathbf{x}(t)} \quad (\text{A.7})$$

Cet estimateur est non-biaisé et convergent. En revanche, il se calcule par une procédure itérative :

$$\begin{cases} \hat{\Sigma}(i+1) = \frac{C}{N_t} \sum_{t=1}^T \frac{\mathbf{x}(t)\mathbf{x}(t)^T}{\mathbf{x}(t)^T \hat{\Sigma}(i)^{-1} \mathbf{x}(t)} \\ \hat{\Sigma}(0) = \hat{\Sigma}_{NSCM} \end{cases} \quad (\text{A.8})$$

Trace Normalized Sample Covariance Matrix (TNSCM)

Cet estimateur est principalement utilisé en BCI dans le cadre de la CSP. Il est équivalent à l'estimateur SCM, à un facteur près.

$$\hat{\Sigma} = \frac{\mathbf{X}\mathbf{X}^T}{\text{Tr}(\mathbf{X}\mathbf{X}^T)} \quad (\text{A.9})$$

Shrinkage Covariance matrix (ShCov)

Cette estimation se base sur une opération de régularisation par *shrinkage* :

$$\hat{\Sigma} = (1 - \gamma)\Sigma_{SCM} + \gamma\nu\mathbf{I} \quad (\text{A.10})$$

où $\gamma \in [0 : 1]$ est le paramètre de la régularisation et ν une constante multiplicative, prise comme la moyenne des valeurs propres de Σ_{SCM} [18].

Le paramètre γ optimal est estimé par la technique suivante [111] :

Soit $z_{ij}(t)$ la quantité suivante :

$$z_{ij}(t) = (x_i(t) - \mu_i)(x_j(t) - \mu_j) \quad (\text{A.11})$$

où $x_i(t)$ et μ_i sont les i -èmes éléments des vecteurs $\mathbf{x}(t)$ et μ , respectivement, avec μ l'espérance de $\mathbf{x}(t)$.

Soit s_{ij} un élément de la matrice de covariance Σ_{SCM} . On calcule le paramètre γ^* optimal par :

$$\gamma^* = \frac{N_t}{(N_t - 1)^2} \frac{\sum_{i,j=1}^C \text{var}(z_{ij}(t))}{\sum_{i \neq j} s_{ij} + \sum_i (s_{ii} - \nu)} \quad (\text{A.12})$$

Local Temporal Covariance Matrix (LTCov)

L'estimation de la matrice de covariance par SCM peut également prendre la forme suivante :

$$\Sigma = \frac{1}{N_t} \sum_{t=1}^{N_t} (\mathbf{x}(t) - \bar{\mathbf{x}})(\mathbf{x}(t) - \bar{\mathbf{x}})^T = \frac{1}{2N_t^2} \sum_{l=1}^{N_t} \sum_{m=1}^{N_t} (\mathbf{x}(l) - \mathbf{x}(m))(\mathbf{x}(l) - \mathbf{x}(m))^T \quad (\text{A.13})$$

On introduira alors un facteur de pondération de manière à pénaliser les fortes variations entre deux échantillons temporels $\mathbf{x}(l)$ et $\mathbf{x}(m)$:

$$\hat{\Sigma} = \frac{1}{2N_t^2} \sum_{l=1}^{N_t} \sum_{m=1}^{N_t} p_{lm} (\mathbf{x}(l) - \mathbf{x}(m))(\mathbf{x}(l) - \mathbf{x}(m))^T \quad (\text{A.14})$$

avec p_{lm} un coefficient paramétré par le couple (γ, τ) tel que :

$$p_{lm} = \begin{cases} \exp \left[-\frac{\|\mathbf{x}(l) - \mathbf{x}(m)\|^2}{\gamma} \right] & \text{si } |l - m| \leq \tau \\ 0 & \text{sinon} \end{cases} \quad (\text{A.15})$$

La condition $|l - m| \leq \tau$ implique qu'un artefact n'aura d'effet que localement dans le calcul, i.e. il ne biaisera l'estimation que pour les points proches de lui temporellement. Le paramètre γ permet quant à lui de pondérer les fortes variations d'amplitudes. γ a généralement pour valeur un multiple de la variance des signaux (typiquement $2\sigma^2$), et τ représente quelques échantillons (typiquement 5).

Minimum Determinant Covariance Matrix (MCD)

La méthode MCD est une méthode d'estimation de matrice de covariance très populaire car elle a un point de rupture élevé ($\alpha = 0.5$) [108]. Cette méthode cherche le sous ensemble d'observations H parmi l'ensemble des observations tel que la matrice de covariance calculée sur ce sous-ensemble soit de déterminant minimum. Cette méthode a deux avantages, elle est très robuste et elle permet de détecter les points aberrants. En effet, les observations ne faisant pas partie du sous ensemble H retenu peuvent être considérées comme faisant dévier le calcul de la matrice de covariance et sont donc assimilées à des points aberrants. En revanche cette méthode a un défaut majeur : elle est très gourmande en calcul. De manière plus détaillée, la recherche du sous-ensemble H se base sur le théorème suivant [108] :

Théorème 1 Soit un jeu de donnée $E = \{\mathbf{x}(1), \dots, \mathbf{x}(N_t)\}$ de N_t observations de C variables. Soit $H_1 \subset \{1, \dots, N_t\}$ avec $|H_1| = h$, posons

$$\bar{\mathbf{x}}_1 = \frac{1}{h} \sum_{t \in H_1} \mathbf{x}(t)$$

$$\boldsymbol{\Sigma}_1 = \frac{1}{h} \sum_{t \in H_1} (\mathbf{x}(t) - \bar{\mathbf{x}}_1)(\mathbf{x}(t) - \bar{\mathbf{x}}_1)^T$$

Si $\det(\boldsymbol{\Sigma}_1) \neq 0$, on définit les distances relatives par :

$$\forall i \in [1 : N_t] \quad d_1(t) = \sqrt{(\mathbf{x}(t) - \bar{\mathbf{x}}_1) \boldsymbol{\Sigma}_1^{-1} (\mathbf{x}(t) - \bar{\mathbf{x}}_1)}$$

Maintenant prenons H_2 tel que $\{d_1(t); t \in H_2\} = \{(d_1)_{1:N_t}, \dots, (d_1)_{h:N_t}\}$ où $(d_1)_{1:N_t} \leq (d_1)_{2:N_t} \leq \dots \leq (d_1)_{N_t:N_t}$ sont les distances ordonnées, calculons $\bar{\mathbf{x}}_2$ et $\boldsymbol{\Sigma}_2$ à partir de H_2 alors :

$$\det(\boldsymbol{\Sigma}_2) \leq \det(\boldsymbol{\Sigma}_1)$$

avec l'égalité si et seulement si $\bar{\mathbf{x}}_1 = \bar{\mathbf{x}}_2$ et $\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2$.

En pratique, le sous-ensemble d'observations H dont la matrice de covariance est de déterminant minimum est obtenu en tirant au hasard un grand nombre de sous-ensemble (par exemple 500) et en appliquant sur eux le théorème 1. Alors parmi ce grand nombre de sous-ensembles, certains convergent plus vite que d'autre vers H . On continue alors le processus uniquement pour les sous-ensembles de convergence rapide.

Finalement, cette méthode repose sur le calcul d'une distance de Mahalanobis. Pour chaque observation $\mathbf{x}(t)$ appartenant à un ensemble d'observation E de moyenne $\bar{\mathbf{x}}$ et de matrice de covariance $\boldsymbol{\Sigma}$, on définit la distance de Mahalanobis $d(t)$ de cette observation par :

$$d(t) = \sqrt{(\mathbf{x}(t) - \bar{\mathbf{x}}) \boldsymbol{\Sigma}^{-1} (\mathbf{x}(t) - \bar{\mathbf{x}})}$$

La distance de Mahalanobis est souvent utilisée pour la détection de données aberrantes et outliers dans un jeu de données. D'une manière très réductrice, on peut dire que l'on construit le sous-ensemble H en éjectant les observations dont la distance de Mahalanobis est grande par rapport aux autres, c'est-à-dire en éjectant les observations divergentes par rapport aux autres.

Comparaison et résultats

La table A.1 donne les résultats en pourcentage de classification correcte pour les différentes méthodes d'estimation des matrices de covariance sur une chaîne de traitement de référence (CSP multi-classe) et pour les 3 jeux de données présentés dans la section 4.4.5.

	User	SCM	TNSCM	NSCM	FP	SHCOV	LTCOV
IV-IIa	U1	77.8	79.8	77.6	78.6	77.3	77.4
	U2	44.4	42.1	43.1	46.2	46.2	46.9
	U3	82.2	82.3	84.1	80.3	82.4	82.5
	U4	58.0	52.8	47.8	49.5	56.0	53.7
	U5	41.0	41.0	41.3	38.2	38.9	41.6
	U6	49.2	48.8	46.6	49.8	48.1	48.6
	U7	80.6	77.6	78.3	76.7	79.4	77.8
	U8	71.2	78.3	80.3	81.4	73.3	71.1
	U9	79.4	77.1	77.6	76.9	73.7	75.0
	moyenne	64.8	64.4	64.1	64.2	63.9	63.8
III-IIIa	k3	91.1	90.3	90.0	89.7	88.3	91.7
	k6	44.2	31.7	34.2	32.1	47.1	46.7
	l1	81.7	79.2	78.3	77.1	81.3	80.0
	moyenne	72.3	67	67.5	66.3	72.2	72.8
Laboratoire	UA	75.0	65.0	55.0	66.7	76.7	71.7
	UB	61.7	66.7	68.3	60.0	63.3	53.3
	UE	63.3	73.3	76.7	73.3	58.3	76.7
	UG	51.7	56.7	50.0	46.7	46.7	51.7
	UH	45.0	40.0	51.7	45.0	41.7	40.0
	UI	81.7	81.7	81.7	81.7	81.7	80.0
	UJ	68.3	75.0	71.7	70.0	73.3	65.0
	UL	63.3	51.7	45.0	51.7	56.7	53.3
	moyenne	63.7	63.7	62.5	61.9	62.3	61.5
	std	15.6	17.4	17.6	17.2	15.9	15.8

TABLE A.1 – Performances en classification pour les différentes méthodes d'estimation sur une chaîne de traitement de référence (CSP multi-classe)

Globalement, nous n'observons pas de différences significatives dans les performances, si ce n'est sur le jeu de données III-IIIa où les méthodes TNSCM, NSCM et FP semblent avoir plus de difficultés à estimer correctement les matrices. Au regard de ces résultats, l'estimateur SCM semble parfaitement adapté à notre problématique.

Common Spatial Pattern

B.1 Approche historique de la CSP

Le calcul des filtres spatiaux par CSP a été introduit dans [104] de la manière suivante : Soit $\Sigma_i \in \mathbb{R}^{C \times C}$ la matrice de covariance estimée sur le test \mathbf{X}_i de classe $y_i \in \{1, 2\}$. On calcule la matrice de covariance moyenne¹ pour chaque classe $\Sigma^{(1)}$ et $\Sigma^{(2)}$:

$$\Sigma^{(k)} = \frac{1}{|\mathcal{I}^{(k)}|} \sum_{i \in \mathcal{I}^{(k)}} \Sigma_i \quad (\text{B.1})$$

avec $\mathcal{I}^{(k)}$ les indices des tests de classe $y_i = k$. On a également la matrice de covariance de l'ensemble des tests :

$$\Sigma^{(*)} = \Sigma^{(1)} + \Sigma^{(2)} \quad (\text{B.2})$$

On effectue premièrement une opération de blanchiment sur $\Sigma^{(*)}$ en déterminant la matrice $\mathbf{U} \in \mathbb{R}^{C \times C}$ telle que :

$$\mathbf{U}^T \Sigma^{(*)} \mathbf{U} = \mathbf{I}_C \quad (\text{B.3})$$

On obtient ainsi les matrices de covariance moyennes intra-classe blanchies :

$$\mathbf{U}^T \Sigma^{(1)} \mathbf{U} = \mathbf{S}^{(1)} \quad \text{et} \quad \mathbf{U}^T \Sigma^{(2)} \mathbf{U} = \mathbf{S}^{(2)} \quad (\text{B.4})$$

On cherche ensuite le projecteur $\mathbf{B} \in \mathbb{R}^{C \times C}$ qui diagonalise conjointement ces deux matrices :

$$\begin{cases} \mathbf{B}^T \mathbf{S}^{(1)} \mathbf{B} = \mathbf{D}^{(1)} = \mathbf{B}^T \mathbf{U}^T \Sigma^{(1)} \mathbf{U} \mathbf{B} \\ \mathbf{B}^T \mathbf{S}^{(2)} \mathbf{B} = \mathbf{D}^{(2)} = \mathbf{B}^T \mathbf{U}^T \Sigma^{(2)} \mathbf{U} \mathbf{B} \end{cases} \quad (\text{B.5})$$

avec $\mathbf{D}^{(k)} = \text{diag}(\{\lambda_c^{(k)}\})$ une matrice diagonale (correspondant à la classe k) et $\lambda_c^{(k)}$ le c -ième élément de cette matrice diagonale, valeur propre de la matrice $\mathbf{S}^{(k)}$. On obtient alors les filtres spatiaux CSP :

$$\boxed{\mathbf{W} = \mathbf{U} \mathbf{B}} \quad (\text{B.6})$$

avec $\mathbf{W} \in \mathbb{R}^{C \times C}$ une matrice également orthogonale.

Grâce à l'opération de blanchiment équation (B.3), on a l'égalité $\mathbf{D}^{(1)} + \mathbf{D}^{(2)} = \mathbf{I}_C$, les valeurs propres $\lambda_c^{(k)}$ sont donc comprises entre 0 et 1. Ces valeurs propres représentent la valeur de la variance moyenne des signaux filtrés spatialement. Ainsi, si la variance du signal filtré de la classe (1) est élevée, elle sera automatiquement faible pour un test de la classe (2), et inversement.

Finalement, les filtres spatiaux sont donc calculés par la résolution d'un problème de diagonalisation jointe sous contrainte d'unicité :

$$\boxed{\begin{cases} \mathbf{W}^T \Sigma^{(1)} \mathbf{W} = \mathbf{D}^{(1)} \\ \mathbf{W}^T \Sigma^{(2)} \mathbf{W} = \mathbf{D}^{(2)} \\ \text{s.c.} \quad \mathbf{D}^{(1)} + \mathbf{D}^{(2)} = \mathbf{I}_C \end{cases}} \quad (\text{B.7})$$

1. Dans le cas présent, c'est la moyenne arithmétique qui est utilisée ; Nous verrons par la suite d'autres manières de calculer la matrice moyenne d'un ensemble de matrices de covariance.

B.2 CSP robuste et régularisée

On distingue principalement deux approches qui permettent de rendre plus robuste le filtrage spatial par CSP. La première joue sur l'estimation des matrices de covariance (cf annexe A), et la seconde sur le calcul numérique des filtres spatiaux.

Estimation robuste des matrices de covariance

La SCM classiquement utilisée pour estimer les matrices de covariance en CSP est connue pour être très sensible aux artefacts et aux points aberrants [55]. Une mauvaise estimation de la matrice de covariance peut alors faire dévier le calcul des filtres spatiaux. En particulier, les artefacts de très grandes amplitudes, relativement courants en EEG, vont modifier le spectre (l'ensemble des valeurs propres) de la matrice de covariance. La valeur propre principale (de plus forte valeur) va être complètement sur-évaluée et lors de la résolution de la CSP, les filtres qui apparaîtront comme les plus discriminants seront en fait biaisés par cet artefact. A titre d'exemple, la figure B.1 représente un ensemble de signaux EEG corrompus par un artefact de forte amplitude (à $t = 160s$).

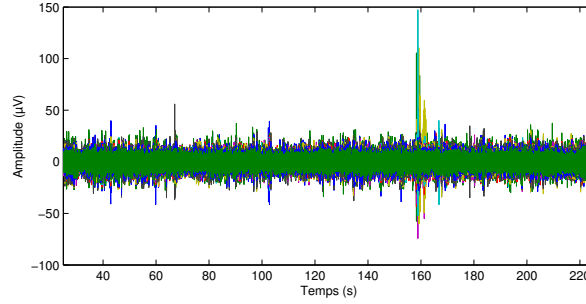


FIGURE B.1 – Signal EEG corrompu par un artefact de forte amplitude

En calculant les filtres spatiaux par CSP sur un tel signal (paradigme mouvement de la main gauche vs. repos) on obtient les caractéristiques présentées figure B.2. On remarque que si la deuxième caractéristique est bien en accord avec les zones motrices de la main, la caractéristique principale est visiblement sans relation avec la tâche mentale en question.

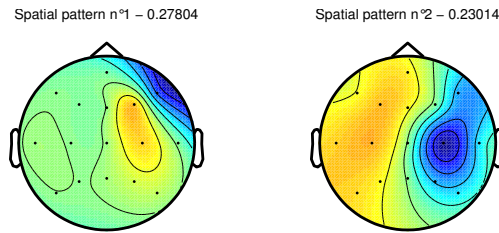


FIGURE B.2 – Caractéristiques spatiales obtenues pour un mouvement de la main gauche vs. repos sur le signal EEG corrompu par un artefact. Le pouvoir discriminant du filtre $|\lambda_c^{(1)} - 0.5|$ est noté dans le titre de la figure.

La figure B.3 présente les caractéristiques spatiales sur le signal nettoyé (suppression manuelle des données EEG contenant l'artefact). On remarque alors que les 2 caractéristiques principales sont bien en accord avec la tâche mentale.

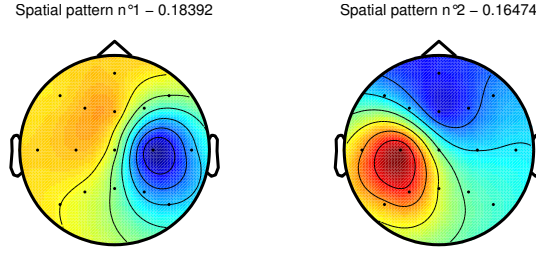


FIGURE B.3 – Caractéristiques spatiales obtenues pour un mouvement de la main gauche vs. repos sur le signal EEG nettoyé manuellement. Le pouvoir discriminant du filtre $|\lambda_c^{(1)} - 0.5|$ est noté dans le titre de la figure.

Si le nettoyage manuel est efficace, une méthode permettant de s'en affranchir est absolument nécessaire pour une utilisation courante d'une ICM. Ainsi des techniques empruntées aux statistiques robustes peuvent être utilisées pour éviter une estimation corrompue des matrices de covariance.

Estimation robuste par MCD : Dans [130, 2], la méthode de *Minimum Covariance Determinant* (MCD) [108] est utilisée pour estimer de manière robuste les matrices de covariance dans le but de calculer des filtres spatiaux par CSP. Cette méthode cherche un sous-ensemble d'observations, parmi l'ensemble des observations, tel que la matrice de covariance calculée sur ce sous-ensemble soit de déterminant minimum. La présentation complète de cet algorithme est disponible en Annexe A. Cette méthode a deux avantages, elle est très robuste et elle permet de détecter les observations aberrantes. En effet les observations ne faisant pas partie du sous-ensemble retenu peuvent être considérées comme faisant dévier le calcul de la matrice de covariance et donc sont assimilées à des points aberrants. La figure B.4 illustre ce principe.

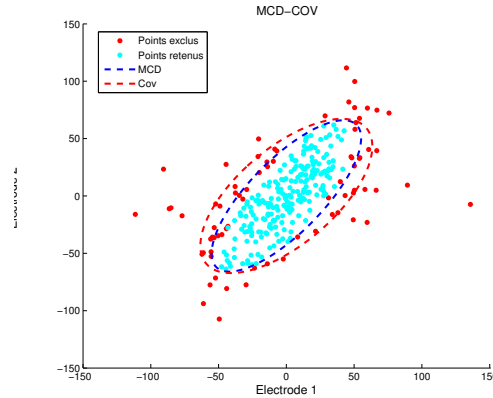


FIGURE B.4 – Observations retenues (en bleu clair) pour l'estimation de la matrice de covariance par MCD. Les points en rouge sont considérés comme étant aberrants.

La MCD est très populaire car elle résiste avec succès à des données corrompues avec plus de 50% de points aberrants [108]. En revanche cette méthode a un défaut majeur : elle est très gourmande en calcul.

Estimation robuste par LTCov : Plutôt que de rejeter les points aberrants comme le fait la MCD, il est possible de pondérer leur impact dans l'estimation de la matrice. Ainsi l'estimation par *Local Temporal Covariance Matrix* joue sur la formulation de l'estimateur pour apporter de la robustesse. Le calcul de cet estimateur est présenté en annexe A. Cet estimateur est appliqué avec succès dans le cadre de la CSP [124].

CSP régularisée

Une autre façon d'apporter de la robustesse au filtrage spatial est de considérer une régularisation qui pénalisera les solutions aberrantes ou encore celles qui s'éloigneraient trop de la réalité physique. Dans [75], une dizaine de solutions se fondant sur ce principe sont recensées et comparées. On distingue deux manières de réaliser cette régularisation, la première s'effectue sur la forme des matrices de covariance, la deuxième sur la fonction de coût à optimiser donnée par l'équation (3.7).

Régularisation des matrices de covariance : Une technique usuelle, dite de *shrinkage*, peut être résumée par l'équation suivante :

$$\hat{\Sigma} = (1 - \alpha)\Sigma + \alpha\mathbf{I}_C \quad (\text{B.8})$$

Le *shrinkage* réalise une interpolation entre la matrice que l'on a estimée par SCM, et la matrice identité. Le but est de rehausser les valeurs propres les plus faibles pour éviter les problèmes numériques d'inversion lors de la résolution de (3.7).

Il est également possible d'appliquer une telle régularisation pour prendre en compte des informations a priori sur les tâches mentales utilisées. Par exemple, dans [76], cette régularisation par shrinkage est doublée d'une régularisation par une matrice de covariance caractéristique² $\bar{\Sigma}$, spécifique au couple de tâche mentale et généralement obtenue par moyennage sur un ensemble de sujets ayant effectué le même paradigme. Cette deuxième régularisation permet de s'affranchir en partie des problèmes expérimentaux qui peuvent survenir pendant l'apprentissage et qui débouchent sur une mauvaise estimation des filtres spatiaux. Cette double régularisation s'exprime alors sous la forme [75] :

$$\hat{\Sigma} = (1 - \alpha) [(1 - \beta)\Sigma + \beta\bar{\Sigma}] + \alpha\mathbf{I}_C \quad (\text{B.9})$$

Ici plus β sera proche de zéro, plus on donnera de poids aux données spécifiques à l'utilisateur, et inversement plus β est proche de 1, plus on restera proche de la matrice de covariance caractéristique.

Régularisation de la fonction de coût : La deuxième manière d'introduire une régularisation est d'appliquer un terme de pénalisation dans la fonction de coût de la CSP donnée par l'équation (3.7) de manière à faire converger les solutions vers des filtres ayant certaines propriétés. Soit $P(\cdot)$ une fonction de $\mathbb{R}^{C \times 1} \rightarrow \mathbb{R}$, le nouveau critère à optimiser prend la forme générale [75] :

$$\frac{\mathbf{w}^T \Sigma^{(1)} \mathbf{w}}{\mathbf{w}^T \Sigma^{(2)} \mathbf{w} + \alpha P(\mathbf{w})} \quad (\text{B.10})$$

où α est un terme de pénalisation réglant l'attache aux données.

La résolution de cette fonction de coût est dépendante de la forme de $P(\cdot)$. Si $P(\cdot)$ est de forme quadratique, $P(\mathbf{w}) = \mathbf{w}^T \mathbf{K} \mathbf{w}$ avec $\mathbf{K} \in \mathbb{R}^{C \times C}$, la résolution s'effectue simplement par diagonalisation de la matrice :

$$\left(\Sigma^{(2)} + \alpha \mathbf{K} \right)^{-1} \Sigma^{(1)} \quad (\text{B.11})$$

2. Cette matrice de covariance caractéristique représente les informations a priori que l'on souhaite prendre en compte

La matrice \mathbf{K} peut alors être construite pour pondérer les fortes variations spatiales des coefficients des filtres pour des électrodes proches [75]. En effet, dans le cas de l'EEG clinique, où on dispose d'une forte densité d'électrode, les signaux de deux électrodes proches sont très corrélés et portent globalement la même information.

Dans [131], une régularisation similaire sur une base de norme \mathcal{L}_1 , c'est-à-dire $P(\mathbf{w}) = \|\mathbf{w}\|_1$, permet d'obtenir des filtres spatiaux parcimonieux, et donc de réaliser par la même occasion une sélection d'électrodes.

Conclusions

L'estimation robuste doit être employée, dans la mesure du possible, de manière systématique. En particulier, il est recommandé d'utiliser une estimation robuste lors des expériences menées dans un cadre de vie courante. En effet, l'estimation robuste ne peut pas dégrader les résultats et sera, dans le pire des cas, équivalente à l'estimation classique par SCM. En revanche, elle introduira une meilleure stabilité lorsque les données seront fortement bruitées.

Le choix de la technique de régularisation à employer se fera en fonction du contexte. Ainsi lorsque le nombre d'électrodes est important, on pourra employer une régularisation de la matrice de covariance par l'intermédiaire de la matrice identité (B.8). Dans le même cas, il est intéressant d'utiliser une régularisation quadratique de la fonction de coût (B.10) qui pénalisera les solutions physiquement aberrantes et dues au nombre important de paramètres dans la résolution de (3.7).

De la même manière, on emploiera une régularisation des matrices de covariance en utilisant une matrice caractéristique (B.9), estimée par exemple à partir des données des autres utilisateurs, lorsque le nombre de données d'apprentissage est faible, ce qui évitera de converger vers des solutions non généralisables. Enfin, l'estimation robuste des matrices de covariance et la régularisation de la CSP peuvent être employées conjointement pour améliorer la robustesse du filtrage spatial.

Séparateur à vaste marge

Introduction

Le SVM repose sur 3 principes :

- Lorsque 2 classes sont linéairement séparables¹, parmi l'ensemble des hyperplans de séparation possible, il existe un hyperplan optimal et cet hyperplan est défini par la maximisation de la distance entre cet hyperplan et les vecteurs les plus proches, appelés vecteurs de support.
- Lorsque 2 classes ne sont pas linéairement séparables, il existe un espace de plus grande dimension (peut-être infinie) dans lequel elles le sont. Trouver cet espace permet alors d'appliquer le premier principe.
- Lorsque 2 classes ne sont pas séparables, on peut alors s'autoriser un certain nombre d'erreurs tout en créant des pénalités pour les vecteurs qui causent cette non-séparabilité.

On a donc deux cas différents à traiter, le cas où les données sont séparables (même si ce n'est pas linéairement, puisque selon le deuxième principe on peut trouver un espace où elles le sont) et le cas où les données sont non séparables.

C.1 Données linéairement séparables

Soit un ensemble de I vecteurs de caractéristiques \mathbf{v}_i de classe $y_i = \{1, -1\}$. On suppose que l'on se trouve dans un espace où les données sont linéairement séparables. Alors l'ensemble des observations de classe 1 se trouvent d'un côté de l'hyperplan séparateur tandis que l'autre ensemble se trouve de l'autre côté de l'hyperplan. Soit un hyperplan paramétré par son vecteur normal \mathbf{w} et un terme de biais b , la distance d'un vecteur de caractéristiques à cet hyperplan s'écrit :

$$D(\mathbf{v}_i, \mathbf{w}, b) = \frac{|\langle \mathbf{w}, \mathbf{v}_i \rangle + b|}{\|\mathbf{w}\|^2} \quad (\text{C.1})$$

Puisque les données sont séparables, on aura $\langle \mathbf{w}, \mathbf{v}_i \rangle + b > 0$ si $y_i = 1$ et $\langle \mathbf{w}, \mathbf{v}_i \rangle + b < 0$ si $y_i = -1$. La distance (C.2) peut donc être reformulée :

$$D(\mathbf{v}_i, \mathbf{w}, b) = \frac{y_i(\langle \mathbf{w}, \mathbf{v}_i \rangle + b)}{\|\mathbf{w}\|^2} \quad (\text{C.2})$$

Finalement on cherche à ce que la distance à l'hyperplan soit maximale pour tous les vecteurs, ce qui garantira que chaque classe soit la plus éloignée possible de l'hyperplan. On introduit donc la notion de marge. Pour tous les vecteurs, la distance d'un vecteur à l'hyperplan est minorée par la distance du vecteur le plus proche à l'hyperplan. Soit s l'indice d'un des vecteurs les plus proches de l'hyperplan, appelé vecteur de support, on a $\forall i, D(\mathbf{v}_i, \mathbf{w}, b) \geq D(\mathbf{v}_s, \mathbf{w}, b)$. Le critère visant à maximiser la marge prend donc la forme suivante :

$$\max_{\mathbf{w}, b} D(\mathbf{v}_s, \mathbf{w}, b) \quad (\text{C.3})$$

1. 2 classes sont linéairement séparables si et seulement si il existe un hyperplan permettant de séparer les deux classes, c'est-à-dire si l'ensemble d'une classe est d'un côté de l'hyperplan tandis que l'autre classe est de l'autre côté.

Étant donné que les vecteurs de support sont les vecteurs les plus proches de l'hyperplan, on a $D(\mathbf{v}_s, \mathbf{w}, b) = \min_{\mathbf{v}_i} D(\mathbf{v}_i, \mathbf{w}, b)$. On reformule donc le problème (C.3) par :

$$\max_{\mathbf{w}, b} \left(\min_{\mathbf{v}_i} \frac{y_i(\langle \mathbf{w}, \mathbf{v}_i \rangle + b)}{\|\mathbf{w}\|^2} \right) \quad (\text{C.4})$$

En ajoutant une contrainte sur \mathbf{w} de manière à ce que la marge soit égale à $\frac{1}{\|\mathbf{w}\|^2}$, alors on reformule le problème (C.4) de manière à obtenir un problème d'optimisation sous contrainte :

$$\begin{cases} \min_{\mathbf{w}, b} \frac{1}{2} \|\mathbf{w}\|^2 \\ \text{s.c.} \quad y_i(\langle \mathbf{w}, \mathbf{v}_i \rangle + b) \geq 1 \quad \forall i \in \{1 : I\} \end{cases} \quad (\text{C.5})$$

La figure C.1 illustre ce principe.

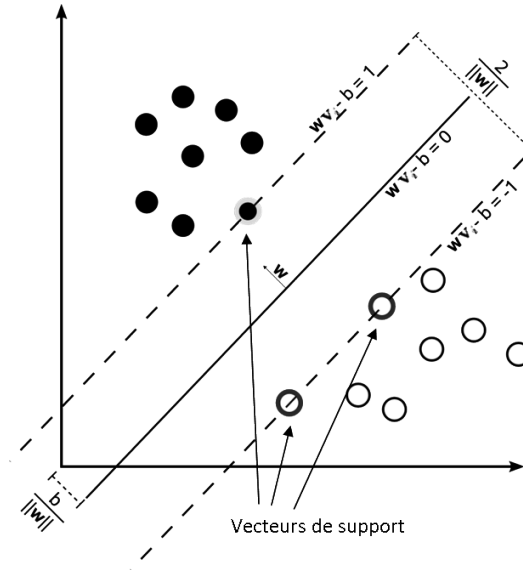


FIGURE C.1 – Vecteurs de support et marge. Le vecteur de projection est défini de manière à ce que la marge soit égale à $\frac{1}{\|\mathbf{w}\|^2}$

Pour résoudre le problème d'optimisation (C.5), on utilise alors le lagrangien suivant :

$$L(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2} \|\mathbf{w}\|^2 + \sum_{i=1}^I \alpha_i (1 - y_i(\langle \mathbf{w}, \mathbf{v}_i \rangle + b)) \quad (\text{C.6})$$

où les α_i sont les multiplicateurs de Lagrange. Le lagrangien doit être minimisé par rapport à \mathbf{w} et b et maximisé par rapport à $\boldsymbol{\alpha}$. Pour minimiser L par rapport à \mathbf{w} et b on annule les dérivées partielles. L'existence et l'unicité du minimum global de L sont assurées par la forme quadratique de L . On obtient alors :

$$\frac{\partial}{\partial \mathbf{w}} L(\mathbf{w}, b, \boldsymbol{\alpha}) = \mathbf{w} - \sum_i \alpha_i y_i \mathbf{v}_i = 0 \Rightarrow \mathbf{w} = \sum_i \alpha_i y_i \mathbf{v}_i \quad (\text{C.7})$$

et

$$\frac{\partial}{\partial b} L(\mathbf{w}, b, \boldsymbol{\alpha}) = \sum_i \alpha_i y_i = 0 \quad (\text{C.8})$$

En injectant (C.7) dans (C.6) on obtient le lagrangien sous sa forme duale :

$$W(\boldsymbol{\alpha}) = \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \langle \mathbf{v}_i, \mathbf{v}_j \rangle \quad (\text{C.9})$$

Le problème d'optimisation (C.5) se transforme donc en :

$$\begin{cases} \max_{\boldsymbol{\alpha}} \left(\sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \langle \mathbf{v}_i, \mathbf{v}_j \rangle \right) \\ \text{s.c.} \quad \alpha_i \geq 0 \quad \forall i \in \{1 : I\} \\ \text{s.c.} \quad \sum_i y_i \alpha_i = 0 \end{cases} \quad (\text{C.10})$$

Le problème (C.5) peut être résolu sans passer par sa forme duale (C.10). Cependant le passage sous la forme duale permet d'explicitier les contraintes de manière plus simple. Il permet également d'introduire directement le produit scalaire entre les \mathbf{v}_i . La résolution de ce problème nous donne la valeur des multiplicateurs de Lagrange qui vont permettre le calcul de \mathbf{w} et b grâce au système suivant :

$$\begin{cases} \mathbf{w}^* = \sum_i \alpha_i y_i \mathbf{v}_i \\ b^* = \sum_i \alpha_i y_i \langle \mathbf{v}_i, \mathbf{v}_s \rangle - y_s \end{cases} \quad (\text{C.11})$$

où s est l'indice d'un des vecteurs de support. En pratique, les α_i sont non nuls seulement pour les vecteurs de support. Ceci est dû au fait que la contrainte $y_i(\langle \mathbf{w}, \mathbf{v}_i \rangle + b) \geq 1$ n'est active que pour les vecteurs de support puisqu'elle est toujours vérifiée pour les autres vecteurs. On se retrouve donc dans le cas où la solution du problème (C.5) n'est dépendante que des vecteurs de support c'est-à-dire d'un sous-ensemble de l'ensemble des observations. Finalement, pour connaître la classe d'une réalisation, il faut regarder de quel côté de l'hyperplan le vecteur de caractéristique correspondant se trouve :

$$y = \text{sign}(\langle \mathbf{w}^*, \mathbf{v} \rangle + b^*) = \text{sign}\left(\sum_i \alpha_i y_i \langle \mathbf{v}_i, \mathbf{v} \rangle + b^*\right) \quad (\text{C.12})$$

où \mathbf{v} est une observation de classe y inconnue.

C.2 Données séparables non linéairement

Dans bien des cas, la frontière entre les classes ne peut être modélisée par un hyperplan. Cependant il est possible d'effectuer un changement de variable, généralement non linéaire, qui va projeter les données dans un espace de plus grande dimension où elles seront alors linéairement séparables. La figure C.2 illustre ce principe.

Soit $\phi(\cdot)$ une fonction permettant de passer dans un espace où les données seront linéairement séparables. En appliquant $\phi(\cdot)$ aux vecteurs de caractéristiques \mathbf{v}_i , on obtient un nouveau jeu de données $\{(\phi(\mathbf{v}_i), y_i)\}$ qui sera linéairement séparable.

En pratique $\phi(\cdot)$ n'est pas une fonction évidente à trouver et $\phi(\mathbf{v}_i)$ peut être complexe à calculer. En remarquant que le problème d'optimisation (C.10) et la fonction de classification (C.12) ne dépendent que du produit scalaire, on introduit la fonction noyau $K(\cdot, \cdot)$, qui est le produit scalaire des vecteurs dans le nouvel espace :

$$K(\mathbf{v}_i, \mathbf{v}_j) = \langle \phi(\mathbf{v}_i), \phi(\mathbf{v}_j) \rangle \quad (\text{C.13})$$

Cette fonction permet de nous affranchir totalement de la connaissance de $\phi(\cdot)$ pour se concentrer exclusivement sur $K(\cdot, \cdot)$ qui en général est plus simple à calculer. Ainsi le problème dual devient :

$$\begin{cases} \max_{\boldsymbol{\alpha}} \left(\sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j K(\mathbf{v}_i, \mathbf{v}_j) \right) \\ \text{s.c.} \quad \alpha_i \geq 0 \quad \forall i \in \{1 : I\} \\ \text{s.c.} \quad \sum_i y_i \alpha_i = 0 \end{cases} \quad (\text{C.14})$$

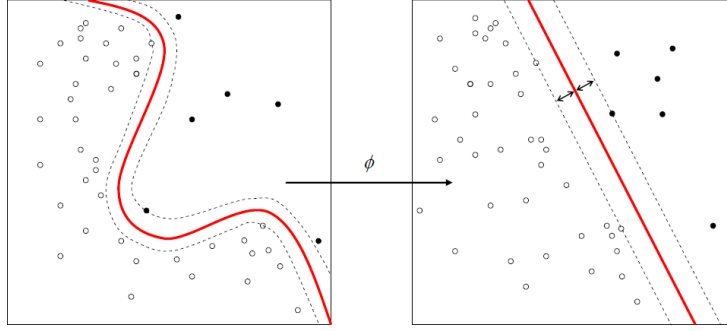


FIGURE C.2 – Lorsque les données sont séparables non linéairement, il est possible de projeter ces données dans un espace où elles le sont. Source de l'image : Wikipédia

et la fonction de classification :

$$y = \text{signe} \left(\sum_i \alpha_i y_i K(\mathbf{v}_i, \mathbf{v}) + b^* \right) \quad (\text{C.15})$$

On pourra alors utiliser comme fonction noyau toute fonction appartenant à l'espace de Hilbert à noyau reproduisant (vérifiant ainsi les propriétés d'un produit scalaire). Une des préoccupations principales en classification SVM est alors de trouver la fonction noyau optimale pour le jeu de données en question. L'une des fonctions noyau la plus utilisée est le noyau gaussien :

$$K(\mathbf{v}_i, \mathbf{v}_j) = \exp(-\gamma \|\mathbf{v}_i - \mathbf{v}_j\|^2) \quad (\text{C.16})$$

où γ est un paramètre du noyau.

C.3 Données non séparables

Lorsque les classes ne sont pas séparables (linéairement ou non), on s'autorise des points mal classés. On modifie ainsi les contraintes en les relâchant grâce à des variables ressort ξ . On parle alors de marge souple. La figure C.3 illustre ce principe.

On introduit alors dans le problème d'optimisation initial (C.5) un terme qui prendra en compte ces points mal classés :

$$\begin{cases} \min_{\mathbf{w}, b} \frac{1}{2} (\|\mathbf{w}\|^2 + C \sum_i \xi_i) \\ \text{s.c.} \quad y_i (\langle \mathbf{w}, \mathbf{v}_i \rangle + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad \forall i \in \{1 : I\} \end{cases} \quad (\text{C.17})$$

où le coefficient C est un réel qui règle le compromis entre la marge possible entre les exemples et le nombre d'erreurs admissibles. Au niveau de la formulation du problème dual, on obtient :

$$\begin{cases} \max_{\alpha} \left(\sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \langle \mathbf{v}_i, \mathbf{v}_j \rangle \right) \\ \text{s.c.} \quad 0 \leq \alpha_i \leq C \quad \forall i \\ \text{s.c.} \quad \sum_i y_i \alpha_i = 0 \end{cases} \quad (\text{C.18})$$

C borne donc chacun des multiplicateurs de Lagrange. La résolution de ce problème est identique à celui développé précédemment, à la différence que l'on force à C tout les α_i supérieur à C . C est généralement réglé par une procédure de validation croisée.

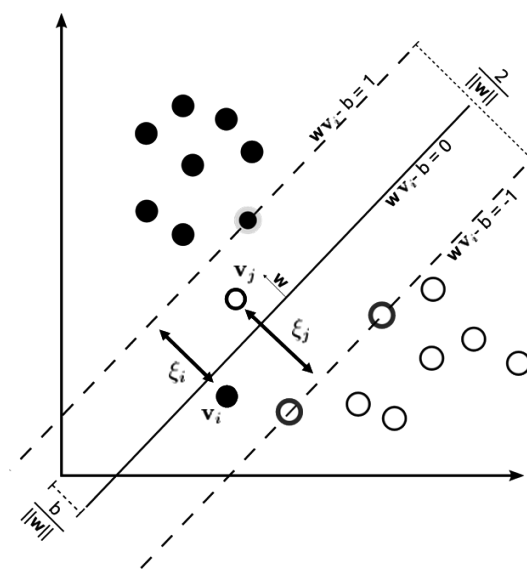


FIGURE C.3 – Cas où les données ne sont pas séparables.

Géométrie de l'information.

D.1 Géodésique

En géométrie Riemannienne, il existe une classe de géodésiques, nommées **affines**, s'il existe une connexion affine dans la variété qui permet de connecter deux plans tangents voisins. Ces géodésiques affines correspondent à des lignes droites dans l'espace tangent, et représentent donc le plus court chemin entre deux points. Pour une paramétrisation telle que $\Gamma(0) = \Sigma_1$ et $\dot{\Gamma}(0) = \mathbf{S}_2$, l'existence et l'unicité d'une telle géodésique est assurée ([64], Chap III, Théorème 6.4).

La définition de ces géodésiques affines implique que le déplacement sur ces courbes est un transport parallèle c'est-à-dire que **dans le référentiel du plan tangent** le vecteur tangent à la géodésique est constant [64, 113].

$$\frac{d}{dt} \langle \dot{\Gamma}(t), \dot{\Gamma}(t) \rangle_{\Gamma(t)} = 0 \quad (\text{D.1})$$

En utilisant la condition initiale on obtient :

$$\|\dot{\Gamma}(t)\|_{\Gamma(t)} = \|\dot{\Gamma}(0)\|_{\Gamma(0)} = \|\mathbf{S}_2\|_{\Sigma_1} \quad (\text{D.2})$$

On peut donc directement exprimer la valeur de la longueur de la géodésique grâce à l'équation (4.6) :

$$L(\Gamma) = \|\mathbf{S}_2\|_{\Sigma_1}$$

En revanche, ne connaissant pas l'expression du vecteur tangent, cela n'est pas suffisant pour calculer cette longueur. On cherche donc à expliciter la formule de la géodésique. On a donc :

$$\|\dot{\Gamma}(t)\|_{\Gamma(t)} = \|\mathbf{S}_2\|_{\Sigma_1} = \|\Sigma_1^{-\frac{1}{2}} \mathbf{S}_2 \Sigma_1^{-\frac{1}{2}}\|_F = \|\tilde{\mathbf{S}}_2\|_F \quad (\text{D.3})$$

en introduisant la notation $\tilde{\mathbf{S}}_2 = \Sigma_1^{-\frac{1}{2}} \mathbf{S}_2 \Sigma_1^{-\frac{1}{2}}$.

On remarque que $\tilde{\mathbf{S}}_2$ est la projection congruente de \mathbf{S}_2 par $\mathbf{W} = \Sigma_1^{-\frac{1}{2}}$. En utilisant les propriétés d'invariance d'une telle projection (cf équation (4.15)) on a $L(\Gamma) = L(\tilde{\Gamma})$ et donc :

$$L(\Gamma) = \int_0^1 \|\tilde{\mathbf{S}}_2\|_F dt = \int_0^1 \sqrt{\text{Tr}(\tilde{\mathbf{S}}_2 \tilde{\mathbf{S}}_2)} dt = \int_0^1 \sqrt{\text{Tr}(\tilde{\mathbf{S}}_2 \tilde{\Gamma}(t) \tilde{\Gamma}^{-1}(t) \tilde{\mathbf{S}}_2 \tilde{\Gamma}(t) \tilde{\Gamma}^{-1}(t))} dt \quad (\text{D.4})$$

Alors par identification avec (4.16) on a :

$$\dot{\tilde{\Gamma}}(t) = \tilde{\mathbf{S}}_2 \tilde{\Gamma}(t) \quad (\text{D.5})$$

qui est une équation différentielle d'ordre 1 admettant comme solution :

$$\tilde{\Gamma}(t) = \exp(t \tilde{\mathbf{S}}_2) \quad (\text{D.6})$$

avec $\exp(\cdot)$ l'opérateur exponentiel matriciel. Soit en retro-projetant par $\Sigma_1^{-\frac{1}{2}}$:

$$\Gamma(t) = \Sigma_1^{\frac{1}{2}} \exp\left(t \Sigma_1^{-\frac{1}{2}} \mathbf{S}_2 \Sigma_1^{-\frac{1}{2}}\right) \Sigma_1^{\frac{1}{2}} \quad (\text{D.7})$$

Par définition $\Gamma(1) = \Sigma_2$, ce qui nous permet d'expliciter la formule du vecteur tangent :

$$\mathbf{S}_2 = \Sigma_1^{\frac{1}{2}} \log \left(\Sigma_1^{-\frac{1}{2}} \Sigma_2 \Sigma_1^{-\frac{1}{2}} \right) \Sigma_1^{\frac{1}{2}} \quad (\text{D.8})$$

En réinjectant (D.8) dans (D.7) on obtient la formule de la géodésique sous sa forme finale :

$$\Gamma(t) = \Sigma_1^{\frac{1}{2}} \left(\Sigma_1^{-\frac{1}{2}} \Sigma_2 \Sigma_1^{-\frac{1}{2}} \right)^t \Sigma_1^{\frac{1}{2}} \quad (\text{D.9})$$

D.2 Moyenne géométrique

Moyenne Riemannienne (géométrique) En utilisant la distance Riemannienne (4.18), et par analogie avec la moyenne Euclidienne, on définit la moyenne Riemannienne de $I \geq 1$ matrices SPD par :

$$\mathfrak{G}(\Sigma_1, \dots, \Sigma_I) = \underset{\Sigma \in P(n)}{\operatorname{argmin}} \sum_{i=1}^I \delta_R^2(\Sigma, \Sigma_i) \quad (\text{D.10})$$

Cette moyenne est également appelée barycentre de Frechet ou encore **moyenne géométrique**. En effet si on considère des matrices de tailles 1×1 tel que $\{x_i > 0\}_{1 \leq i \leq I}$, cette définition donne $\mathfrak{G}(x_1, \dots, x_I) = \operatorname{argmin}_{x > 0} \sum_{i=1}^I \log^2(x_i/x) = \sqrt[I]{x_1 \dots x_I}$.

Pour une variété à courbure négative comme $P(n)$, un tel minimum local existe et il est unique [62]. Cependant il n'y a pas de formule littérale pour calculer cette moyenne et une procédure d'optimisation doit être employée.

Il existe plusieurs façons de réaliser cette optimisation. Dans [13], F. Barbaresco utilise le fait que lorsque l'on se place au barycentre¹ des I points, le champ de Jacobi normal s'annule. Dans ce cas, la résultante des I vecteurs tangents partant du barycentre vers les I points est nulle. Ce concept est schématisé figure D.1.

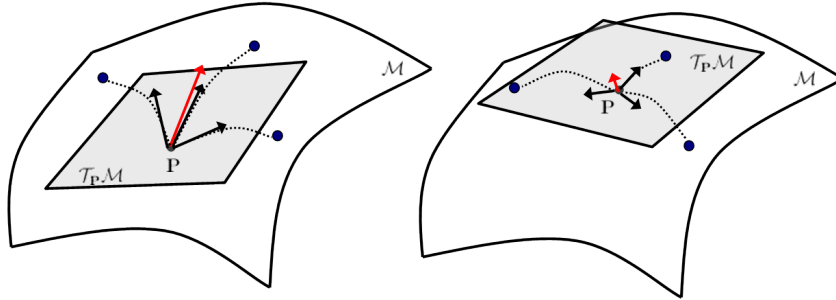


FIGURE D.1 – Calcul de la moyenne géométrique de 3 points. En pointillé, les géodésiques, en trait plein les vecteurs tangents et en rouge la résultante des vecteurs tangents. La résultante donne la direction de l'optimisation et s'annule lorsque le plan tangent est calculé à la moyenne géométrique.

Ainsi on peut exprimer cette résultante à partir de la formulation du vecteur tangent (5.2) :

$$\mathbf{J} = \sum_i^I \Sigma^{\frac{1}{2}} \log \left(\Sigma^{-\frac{1}{2}} \Sigma_i \Sigma^{-\frac{1}{2}} \right) \Sigma^{\frac{1}{2}} \quad (\text{D.11})$$

1. Pour la géométrie Riemannienne, la notion de moyenne est identique à celle de barycentre.

Lorsque Σ est le barycentre, on a $\mathbf{J} = \mathbf{0}$ (une matrice nulle) ce qui implique :

$$\sum_i^I \log \left(\Sigma^{-\frac{1}{2}} \Sigma_i \Sigma^{-\frac{1}{2}} \right) = \mathbf{0} \quad (\text{D.12})$$

En projetant cette résultante dans la variété riemannienne grâce à la carte exponentielle, on obtient la procédure itérative suivante :

$$\Sigma_{(t+1)} = \Sigma_{(t)}^{\frac{1}{2}} \exp \left(\epsilon \sum_i^I \log \left(\Sigma_{(t)}^{-\frac{1}{2}} \Sigma_i \Sigma_{(t)}^{-\frac{1}{2}} \right) \right) \Sigma_{(t)}^{\frac{1}{2}} \quad (\text{D.13})$$

avec t l'indice de la procédure itérative et ϵ un paramètre déterminant la "profondeur" de l'itération. Le processus s'arrête dès que la norme du champ de Jacobi est suffisamment faible, i.e $\|\mathbf{J}\|_F < \xi$. L'initialisation de cette procédure n'est pas critique, mais peut influencer fortement le nombre d'itération nécessaires. On peut considérer la moyenne arithmétique comme une bonne solution à ce problème d'initialisation.

Une autre manière de calculer cette moyenne est de considérer le fait que le plan tangent est localement un homéomorphisme de la variété, et étant euclidien il est possible d'y calculer une moyenne euclidienne pour ensuite la projeter sur la variété. On procède également par itérations successives, consistant à 1. projeter dans le plan tangent l'ensemble des matrices, 2. estimer la moyenne arithmétique des vecteurs du plan tangent, 3. projeter cette moyenne dans la variété, 4. recommencer la procédure avec la nouvelle moyenne comme point de passage vers le plan tangent. La figure D.2 schématise ce processus.

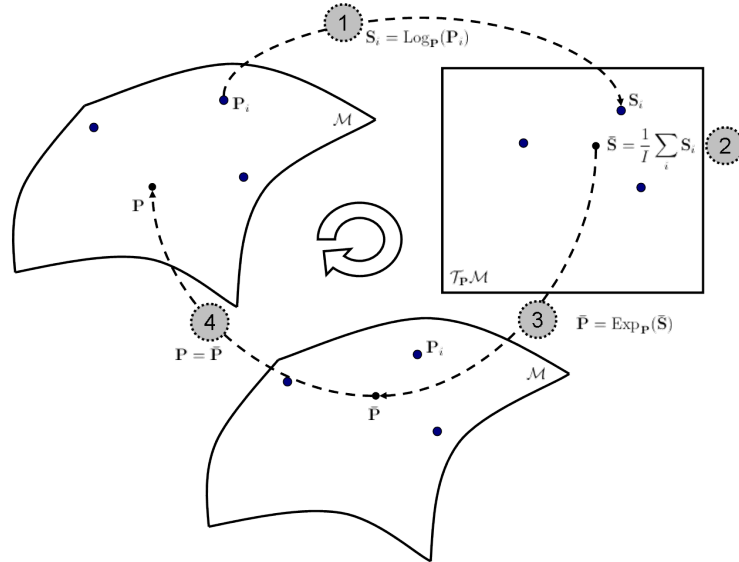


FIGURE D.2 – Processus d'estimation de la moyenne géométrique par estimation de la moyenne arithmétique dans le plan tangent. 1 - On projette les données dans le plan tangent grâce à l'opérateur logarithmique. 2 - Dans le plan tangent, on estime la moyenne arithmétique des données. 3 - On projette dans la variété grâce à l'opérateur exponentiel la moyenne arithmétique. 4 - On recommence la procédure avec la nouvelle estimation.

Numériquement, cette procédure est équivalente à la précédente, avec $\epsilon = \frac{1}{I}$.

D.3 Médiane

La médiane peut être définie comme le point qui minimise la somme des distances [15] :

$$\mathfrak{M}(\Sigma_1, \dots, \Sigma_I) = \underset{\Sigma \in P(n)}{\operatorname{argmin}} \sum_{i=1}^I \delta_R(\Sigma, \Sigma_i) \quad (\text{D.14})$$

Ici aussi il est également possible de passer par une procédure itérative dont la formule est la suivante [12] :

$$\Sigma_{(t+1)} = \Sigma_{(t)}^{\frac{1}{2}} \exp \left(\epsilon \sum_i \frac{\log \left(\Sigma_{(t)}^{-\frac{1}{2}} \Sigma_i \Sigma_{(t)}^{-\frac{1}{2}} \right)}{\|\log \left(\Sigma_{(t)}^{-\frac{1}{2}} \Sigma_i \Sigma_{(t)}^{-\frac{1}{2}} \right)\|_F} \right) \Sigma_{(t)}^{\frac{1}{2}} \quad (\text{D.15})$$

En pratique, on observe des problèmes de convergence avec cette équation de mise à jour. Il est possible d'utiliser une régularisation pour palier à ce problème [12].

Il est également possible (et recommandé) de passer par l'espace tangent pour calculer la médiane, en suivant la même procédure que celle proposée pour la moyenne décrite à la figure D.2.

Distances entre matrices de covariance

La comparaison de matrices de covariance est une problématique ancienne et peut être abordée par un autre prisme que celui de la géométrie de l'information. Ainsi il existe dans la littérature un certain nombre de moyens à notre disposition pour calculer une distance entre deux matrices de covariance. Nous verrons ici une sélection non exhaustive des principales d'entre elles.

E.1 Distance Euclidienne :

La distance Euclidienne est simplement définie à partir de la norme de Frobenius.

$$\delta_E(\Sigma_1, \Sigma_2) = \|\Sigma_1 - \Sigma_2\|_F = \sqrt{\text{Tr}((\Sigma_1 - \Sigma_2)(\Sigma_1 - \Sigma_2)^T)} \quad (\text{E.1})$$

Cette distance revient à vectoriser les matrices et à calculer la distance euclidienne entre les deux vecteurs suivant la norme \mathcal{L}_2 . Elle ne tient donc pas compte des relations qui lient les éléments d'une matrice de covariance entre eux, c'est pourquoi elle est relativement peu efficace pour quantifier une distance entre matrices de covariance.

Moyenne Euclidienne (Arithmétique) En utilisant la distance Euclidienne entre les matrices de covariance, il est possible de définir la moyenne de $I \geq 1$ matrices SPD par :

$$\mathfrak{A}(\Sigma_1, \dots, \Sigma_I) = \underset{\Sigma \in P(n)}{\text{argmin}} \sum_{i=1}^I \delta_E^2(\Sigma, \Sigma_i) = \frac{1}{I} \sum_{i=1}^I \Sigma_i \quad (\text{E.2})$$

Cette moyenne est également connue sous le nom de moyenne arithmétique [85]. Cette moyenne est couramment utilisée en CSP.

E.2 Divergence et distance de Kullback-Leibler

La divergence de Kullback-Leibler permet de mesurer la dissimilarité entre deux distributions. Soit P et Q deux distributions de probabilité continues, la divergence de Kullback-Leibler est définie par :

$$D_{KL}(P\|Q) = \int_{-\infty}^{+\infty} p(x) \log \frac{p(x)}{q(x)} dx \quad (\text{E.3})$$

Lorsque les distributions suivent des lois normales multivariées (de dimension n) centrées $\mathcal{N}_1(0, \Sigma_1)$ et $\mathcal{N}_2(0, \Sigma_2)$, la divergence prend la forme suivante :

$$D_{KL}(\mathcal{N}_1\|\mathcal{N}_2) = \frac{1}{2} \left(\text{Tr}(\Sigma_2^{-1}\Sigma_1) - \log \left(\frac{\det \Sigma_1}{\det \Sigma_2} \right) - n \right) \quad (\text{E.4})$$

Le calcul de la divergence entre ces deux distributions se base donc entièrement sur les matrices de covariance. Cette divergence n'est pas commutative, on introduit la distance de Kullback-Leibler symétrique pour pallier à ce problème :

$$\delta_{KL}(P, Q) = \sqrt{D_{KL}(P\|Q) + D_{KL}(Q\|P)} \quad (\text{E.5})$$

Toujours dans le cadre de lois normales centrées, on obtient :

$$\delta_{KL}(\Sigma_1, \Sigma_2) = \sqrt{\frac{1}{2} \text{Tr}(\Sigma_1 \Sigma_2^{-1} + \Sigma_1^{-1} \Sigma_2 - 2\mathbf{I}_n)} \quad (\text{E.6})$$

En réalité, cette distance n'en est pas vraiment une, car elle ne respecte pas l'inégalité triangulaire¹. On notera également que la divergence de Kullback-Leibler est liée à la courbure de la variété Riemannienne des distributions de probabilité. Ainsi on s'attend à obtenir des résultats similaires à ceux obtenus par l'intermédiaire de la distance Riemannienne.

E.3 Distance Log-Euclidienne

Si la métrique Riemannienne définie par l'équation (4.14) est considérée comme la métrique naturelle au sens de la géométrie de l'information, elle n'est pas unique d'un point de vue mathématique. En particulier, toute fonction satisfaisant les propriétés d'un produit scalaire dans l'espace tangent peut être considérée comme une métrique viable pour la variété en question.

On peut alors définir d'autres métriques en fonction des propriétés que l'on souhaite fournir à la distance qui en émane. Une métrique alternative partant des propriétés de bi-invariance² et d'invariance par similarité³ est définie dans [4]. Elle entraîne la définition de la distance dite Log-Euclidienne :

$$\delta_{LE}(\Sigma_1, \Sigma_2) = \|\log(\Sigma_2) - \log(\Sigma_1)\|_F = \delta_E(\log(\Sigma_1), \log(\Sigma_2)) \quad (\text{E.7})$$

qui finalement se résume à la distance euclidienne du logarithme des deux matrices. Dans le cas où deux matrices SPD \mathbf{A} et \mathbf{B} commutent, on a $\log(\mathbf{AB}) = \log(\mathbf{A}) + \log(\mathbf{B})$. La distance Log-Euclidienne est donc égale à la distance Riemannienne lorsque les deux matrices considérées commutent.

On remarquera en particulier que pour toute matrice SPD on a l'égalité suivante :

$$\delta_R(\Sigma_1, \Sigma_2) = \delta_{LE}(\mathbf{I}_n, \Sigma_1^{-\frac{1}{2}} \Sigma_2 \Sigma_1^{-\frac{1}{2}}) \quad (\text{E.8})$$

Cette distance est généralement utilisée comme une approximation de la distance Riemannienne.

Moyenne Log-Euclidienne : En se basant sur le principe décrit à l'équation (E.7), on obtient de manière évidente la moyenne Log-Euclidienne par :

$$\mathfrak{B}(\Sigma_1, \dots, \Sigma_I) = \exp[\mathfrak{A}(\log \Sigma_1, \dots, \log \Sigma_I)] = \exp\left(\frac{1}{I} \sum_{i=1}^I \log(\Sigma_i)\right) \quad (\text{E.9})$$

1. Pour qu'une fonction puisse être qualifiée de distance, elle doit vérifier 3 propriétés : Symétrie $\forall x, y, d(x, y) = d(y, x)$, Séparation $\forall x, y, d(x, y) = 0 \Leftrightarrow x = y$ et l'inégalité triangulaire $\forall x, y, z, d(x, z) \leq d(x, y) + d(y, z)$

2. invariance par multiplication à droite et à gauche, $f(BA) = f(AB) = f(A)$

3. $f(B^{-1}AB) = f(A)$

E.4 Récapitulatif

TABLE E.1 – Comparaison des différentes distances

	Euclidienne	Riemannienne
Distance	$\ \Sigma_2 - \Sigma_1\ _F$	$\ \log\left(\Sigma_1^{-\frac{1}{2}}\Sigma_2\Sigma_1^{-\frac{1}{2}}\right)\ _F$
Géodésique	$(1-t)\Sigma_1 + t\Sigma_2$	$\Sigma_1^{\frac{1}{2}}\left(\Sigma_1^{-\frac{1}{2}}\Sigma_2\Sigma_1^{-\frac{1}{2}}\right)^t\Sigma_1^{\frac{1}{2}}$
Moyenne de 2 pts	$\frac{1}{2}(\Sigma_1 + \Sigma_2)$	$\Sigma_1^{\frac{1}{2}}\left(\Sigma_1^{-\frac{1}{2}}\Sigma_2\Sigma_1^{-\frac{1}{2}}\right)^{\frac{1}{2}}\Sigma_1^{\frac{1}{2}}$
Moyenne de I pts	$\frac{1}{I}\sum_{i=1}^I \Sigma_i$	cf section 4.3.3
	Log Euclidienne	Kullback-Leibler
Distance	$\ \log(\Sigma_2) - \log(\Sigma_1)\ _F$	$\sqrt{\frac{1}{2}\text{Tr}(\Sigma_1\Sigma_2^{-1} + \Sigma_1^{-1}\Sigma_2 - 2\mathbf{I})}$
Géodésique	$\exp((1-t)\log(\Sigma_1) + t\log(\Sigma_2))$	Non définie
Moyenne de 2 pts	$\exp\left(\frac{1}{2}(\log(\Sigma_1) + \log(\Sigma_2))\right)$	Non définie
Moyenne de I pts	$\exp\left(\frac{1}{I}\sum_{i=1}^I \log(\Sigma_i)\right)$	Non définie

Liste des publications personnelles

Journal international

- A. Barachant, S. Bonnet, M. Congedo, C. Jutten, Multi-Class Brain Computer Interface Classification by Riemannian Geometry, *IEEE Trans. On Biomed. Eng.*, Vol 59 (4), pp 920-928, 2012.

Communications avec comité de lecture

- A. Barachant, T. Aksenova, S. Bonnet, Spatial filtering optimisation in motor imagery EEG-based BCI, *NeuroComp 2008*
- A. Barachant, T. Aksenova, S. Bonnet, Filtrage spatial robuste à partir d'un sous-ensemble optimal d'électrodes en BCI EEG, *GRETSI 2009*
- A. Barachant, S. Bonnet, M. Congedo, C. Jutten, Common Spatial Pattern revisited by Riemannian Geometry, *IEEE MMSP 2010*
- A. Barachant, S. Bonnet, M. Congedo, C. Jutten, Riemannian Geometry applied to BCI classification, *LVA/ICA 2010*
- A. Barachant, S. Bonnet, M. Congedo, C. Jutten, A Brain-Switch Using Riemannian Geometry, in *Proceedings of the 5th international Brain Computer Interface conference, Graz University of Technology, Austria, 2011.*
- A. Barachant, S. Bonnet, Channel selection procedure using riemannian distance for BCI applications, *5th International IEEE/EMBS Conference on Neural Engineering (NER), 2011.*
- A. Barachant, S. Bonnet, M. Congedo, C. Jutten, Réalisation d'un Brain-Switch EEG par Géométrie Riemannienne, *GRETSI 2011.*

Communication sans comité de lecture

- A. Barachant, Mise en œuvre d'un Brain Switch EEG, BCI Meeting à Paris, Mai 2010

Brevet

- A. Barachant, S. Bonnet, Procédé et système de classification de signaux neuronaux, et procédé de sélection d'électrodes pour interface neuronale direct, 2010

Bibliographie

- [1] H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control.*, 19(6) :716–723, 1974.
- [2] T. Aksenova, A. Barachant, and S. Bonnet. Spatial filtering optimisation in motor imagery EEG-based BCI. In *2ème Conférence Française des Neurosciences Computationnelles - NeuroComp*, pages 213–216, 2008.
- [3] K. K. Ang, Z.Y. Chin, H. Zhang, and G. Cuntai. Filter bank common spatial pattern (FBCSP) in Brain-Computer interface. In *IEEE International Joint Conference on Neural Networks*, pages 2390–2397, 2008.
- [4] V. Arsigny, P. Fillard, X. Pennec, and N. Ayache. Geometric means in a novel vector space structure on symmetric Positive-Definite matrices. *SIAM Journal on Matrix Analysis and Applications*, 29(1) :328–347, January 2007.
- [5] A. Barachant, T. Aksenova, and S. Bonnet. Filtrage spatial robuste à partir d’un sous-ensemble optimal d’électrodes en BCI EEG. In *22ème colloque GRETSI*, 2009.
- [6] A. Barachant and S. Bonnet. Channel selection procedure using riemannian distance for BCI applications. In *2011 5th International IEEE/EMBS Conference on Neural Engineering (NER)*, pages 348–351. IEEE, 2011.
- [7] A. Barachant, S. Bonnet, M. Congedo, and C. Jutten. Common spatial pattern revisited by riemannian geometry. In *IEEE International Workshop on Multimedia Signal Processing (MMSP)*, pages 472–476, 2010.
- [8] A. Barachant, S. Bonnet, M. Congedo, and C. Jutten. Riemannian geometry applied to BCI classification. In *9th International Conference Latent Variable Analysis and Signal Separation LVA/ICA*, volume 6365 LNCS, pages 629–636, 2010.
- [9] A. Barachant, S. Bonnet, M. Congedo, and C. Jutten. A Brain-Switch using riemannian geometry. In *Proceedings of the 5th international Brain Computer Interface conference*, pages 64–67, Gratz, September 2011. Graz University of Technology, Austria.
- [10] A. Barachant, S. Bonnet, M. Congedo, and C. Jutten. Multi-class brain computer interface classification by riemannian geometry. *IEEE Transactions on Bio-Medical Engineering*, 2011.
- [11] A. Barachant, S. Bonnet, M. Congedo, and C. Jutten. Réalisation d’un Brain-Switch EEG par géométrie riemannienne. In *23ème colloque GRETSI*, Bordeaux, France, 2011.
- [12] F. Barbaresco. Géométrie, médiane et diffusion de spectres de fréquences acoustiques : Ou comment ”médiatiser” ou ”diffuser” des morceaux de musiques.
- [13] F. Barbaresco. Innovative tools for radar signal processing based on cartan’s geometry of SPD matrices & information geometry. In *IEEE Radar Conference*, 2008.
- [14] F. Barbaresco. Interactions between symmetric cone and information geometries : Bruhat-Tits and siegel spaces models for high resolution autoregressive doppler imagery. In *Emerging Trends in Visual Computing : LIX Fall Colloquium, ETVC*, pages 124–163. Springer-Verlag, 2008.
- [15] F. Barbaresco. Géométrie différentielle des matrices de covariance & espaces métriques à courbure négative. In *22ème colloque GRETSI*, 2009.
- [16] Y. Benjamini and Y. Hochberg. Multiple hypotheses testing with weights. *Scandinavian Journal of Statistics*, 24(3) :407–418, 1997.

- [17] G.E. Birch, Z. Bozorgzadeh, and S.G. Mason. Initial on-line evaluations of the LF-ASD brain-computer interface with able-bodied and spinal-cord subjects using imagined voluntary motor potentials. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 10(4) :219–224, 2002.
- [18] B. Blankertz, S. Lemm, M. Treder, S. Haufe, and K.-R. Müller. Single-trial analysis and classification of ERP components - a tutorial. *NeuroImage*, June 2010.
- [19] B. Blankertz, R. Tomioka, S. Lemm, M. Kawanabe, and K.-R. Müller. Optimizing spatial filters for robust EEG Single-Trial analysis. *IEEE Signal Processing Magazine*, 25(1) :41–56, 2008.
- [20] J.F. Borisoff, S.G. Mason, A. Bashashati, and G.E. Birch. Brain-computer interface design for asynchronous control applications : improvements to the LF-ASD asynchronous brain switch. *Biomedical Engineering, IEEE Transactions on*, 51(6) :985–992, 2004.
- [21] Z. Bozorgzadeh, G.E. Birch, and S.G. Mason. The LF-ASD brain computer interface : on-line identification of imagined finger flexions in the spontaneous EEG of able-bodied subjects. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 6, pages 2385–2388, 2000.
- [22] R. Caraballona and P. Castiglioni. To feedback or not to feedback ? some open questions for brain-computer interfaces use. In *IFMBE Proceedings*, volume 25 IFMBE, pages 309–311, 2009.
- [23] H. Cecotti, B. Rivet, M. Congedo, C. Jutten, O. Bertrand, E. Maby, and J. Mattout. Suboptimal sensor subset evaluation in a p300 Brain-Computer interface. In *18th European Signal Processing Conference (EUSIPCO-2010)*, 2010.
- [24] R. Chavarriaga and J.D.R. Millán. Learning from EEG error-related potentials in noninvasive brain-computer interfaces. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 18(4) :381–388, 2010.
- [25] F. Cincotti, L. Kauhanen, F. Aloise, T. Palomäki, N. Caporusso, P. Jylänki, D. Mattia, F. Babiloni, G. Vanacker, M. Nuttin, M. G. Marciani, and J. d. R. Millan. Vibrotactile feedback for Brain-Computer interface operation. *Computational Intelligence and Neuroscience*, 2007.
- [26] S. Coyle, T. Ward, C. Markham, and G. McDarby. On the suitability of near-infrared (NIR) systems for next-generation brain-computer interfaces. *Physiological Measurement*, 25(4) :815–822, 2004.
- [27] S. M. Coyle, T. E. Ward, and C. M. Markham. Brain-computer interface using a simplified functional near-infrared spectroscopy system. *Journal of Neural Engineering*, 4(3) :219–226, September 2007.
- [28] C. Delmaire, A. Krainik, V. Lethuc, N. Reyns, H. Duffau, L. Capelle, and S. Lehericy. IRM fonctionnelle cérébrale : bases physiologiques, techniques et applications cliniques. *Journal de Radiologie*, 88(3, Part 2) :497–509, March 2007.
- [29] T. Dickhaus, C. Sannelli, K. R. Müller, G. Curio, and B. Blankertz. Predicting BCI performance to study BCI illiteracy. *BMC Neuroscience*, 10(Suppl 1) :P84, July 2009.
- [30] G. Dornhege, B. Blankertz, G. Curio, and K. -R Muller. Boosting bit rates in noninvasive EEG single-trial classifications by feature combination and multiclass paradigms. *IEEE Trans. Biomed. Eng.*, 51(6) :993–1002, 2004.
- [31] R. O. Duda, P. E. Hart, and D.G. Stork. *Pattern Classification*, chapter 3.8.2, pages 117–124. J. Wiley & Sons Inc, 2nd revised edition edition, November 2000.
- [32] E. S. Edgington. A normal curve method for combining probability values from independent experiments. *Journal of Psychology : Interdisciplinary and Applied*, 82(1) :85–89, 1972.

- [33] J. Farquhar. A linear feature space for simultaneous learning of spatio-spectral filters in BCI. *Neural Networks*, 22(9) :1278–1285, November 2009.
- [34] M. Fatourehchi, R K Ward, and G E Birch. A self-paced brain–computer interface system with a low false positive rate. *Journal of Neural Engineering*, 5 :9–23, March 2008.
- [35] M. S. Fernandes, N. S. Dias, A. F. Silva, J. S. Nunes, S. Lanceros-Méndez, J. H. Correia, and P. M. Mendes. Hydrogel-based photonic sensor for a biopotential wearable recording system. *Biosensors and Bioelectronics*, 26(1) :80–86, September 2010.
- [36] A. Ferreira, T.F. Bastos-Filho, M. Sarcinelli-Filho, F.A. Cheein, J.F. Postigo, and R. Carelli. Teleoperation of an industrial manipulator through a TCP/IP channel using EEG signals. In *IEEE International Symposium on Industrial Electronics*, volume 4, pages 3066–3071, 2006.
- [37] P. Ferrez. *Error-related EEG potentials in brain-computer interfaces*. PhD thesis, 2007.
- [38] P. Thomas Fletcher and Sarang Joshi. Principal geodesic analysis on symmetric spaces : Statistics of diffusion tensors. In *Computer Vision and Mathematical Methods in Medical and Biomedical Image Analysis*, pages 87–98, 2004.
- [39] P Thomas Fletcher, Conglin Lu, Stephen M Pizer, and Sarang Joshi. Principal geodesic analysis for the study of nonlinear statistics of shape. *IEEE Transactions on Medical Imaging*, 23(8) :995–1005, August 2004.
- [40] J. Friedman, T. Hastie, and R. Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33, February 2010.
- [41] K. Fukunaga. *Introduction to statistical pattern recognition*. Academic Press, 1990.
- [42] A. Furdea, S. Halder, D.J. Krusienski, D. Bross, F. Nijboer, N. Birbaumer, and A. Kübler. An auditory oddball (P300) spelling system for brain-computer interfaces. *Psychophysiology*, 46(3) :617–625, 2009.
- [43] F. Galán, M. Nuttin, E. Lew, P.W. Ferrez, G. Vanacker, J. Philips, and J. del R. Millán. A brain-actuated wheelchair : Asynchronous and non-invasive brain-computer interfaces for continuous control of robots. *Clinical Neurophysiology*, 119(9) :2159–2169, 2008.
- [44] D. Garrett, D.A. Peterson, C.W. Anderson, and M.H. Thaut. Comparison of linear, nonlinear, and feature selection methods for eeg signal classification. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 11(2) :141–144, 2003.
- [45] T. Geng, M. Dyson, C.S. Tsui, and J.Q. Gan. A 3-class asynchronous BCI controlling a simulated mobile robot. In *29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 2524–2527, 2007.
- [46] T. Geng and J. Q. Gan. Motor prediction in Brain-Computer interfaces for controlling mobile robots. In *30th Annual International IEEE Conference of the Engineering in Medicine and Biology Society*, pages 634–637, 2008.
- [47] C. Gouy-Pailler. *Vers une modélisation dynamique de l’activité cérébrale pour la conception d’interfaces cerveau-machines asynchrones*. These, Université Joseph-Fourier - Grenoble I, 2009.
- [48] C. Gouy-Pailler, M. Congedo, C. Brunner, C. Jutten, and G. Pfurtscheller. Nonstationary brain source separation for multiclass motor imagery. *IEEE Trans. Biomed. Eng.*, 57(2) :469–478, 2010.
- [49] B. Graimann, J. E. Huggins, S. P. Levine, and G. Pfurtscheller. Visualization of significant ERD/ERS patterns in multichannel EEG and ECoG data. *Clinical Neurophysiology*, 113(1) :43–47, January 2002.

- [50] M. Grosse-Wentrup and M. Buss. Multiclass common spatial patterns and information theoretic feature extraction. *IEEE Trans. Biomed. Eng.*, 55(8) :1991–2000, 2008.
- [51] C. S. Herrmann. Human EEG responses to 1-100 hz flicker : resonance phenomena in visual cortex and their potential correlation to cognitive phenomena. *Experimental Brain Research*, 137(3) :346–353, 2001.
- [52] T. Hinterberger, G. Baier, J. Mellinger, and N. Birbaumer. Auditory feedback of human EEG for direct Brain-Computer communication. In *international conference of auditory display (ICAD)*, 10 :1775–1777, 2004.
- [53] A. E. Hoerl and R. W. Kennard. Ridge regression : Biased estimation for nonorthogonal problems. *Technometrics*, 12(1) :55–67, 1970.
- [54] C. Holzner, C. Guger, C. Grönegress, G. Edlinger, and M. Slater. Using a p300 brain computer interface for smart home control. In *IFMBE Proceedings*, pages 174–177, 2009.
- [55] P. J. Huber. *Robust Statistics*. Wiley-Interscience, 1981.
- [56] I. Iturrate, J. Antelis, and J. Minguez. Synchronous EEG brain-actuated wheelchair with automated navigation. In *IEEE International Conference on Robotics and Automation, ICRA*, pages 2318–2325, 2009.
- [57] C. Jia, H. Xu, B. Hong, X. Gao, Z. Zhang, and S. Gao. A human computer interface using SSVEP-based BCI technology. In *Proceedings of the 3rd international conference on Foundations of augmented cognition*, pages 113–119, Beijing, China, 2007.
- [58] W. Jouini, D. Le Guennec, C. Moy, and J. Palicot. Log-normal approximation of chi-square distributions for signal processing. In *Proceedings of General Assembly of URSI*, 2011.
- [59] N. Jrad, M. Congedo, R. Phlypo, S. Rousseau, R. Flamary, F. Yger, and A. Rakotomamonjy. sw-svm : sensor weighting support vector machines for eeg-based brain-computer interfaces. *Journal of Neural Engineering*, 8(5) :056004, 2011.
- [60] T.-P. Jung, S. Makeig, C. Humphries, T.-W. Lee, M.J. Mckeown, V. Iragui, and T.J. Sejnowski. Removing electroencephalographic artifacts by blind source separation. *Psychophysiology*, 37(2) :163–178, 2000.
- [61] M. Kamrunnagar, N.S. Dias, and S.J. Schiff. Optimization of electrode channels in brain computer interfaces. *Annual International Conference of the IEEE Engineering in Medicine and Biology Society.*, 2009 :6477–6480, 2009.
- [62] H. Karcher. Riemannian center of mass and mollifier smoothing. *Communications on Pure and Applied Mathematics*, 30(5) :509–541, 1977.
- [63] S.P. Kelly, E.C. Lalor, C. Finucane, G. McDarby, and R.B. Reilly. Visual spatial attention control in an independent brain-computer interface. *IEEE Trans. Biomed. Eng.*, 52(9) :1588–1596, 2005.
- [64] S. Kobayashi and K. Nomizu. *Foundations of differential geometry*. Wiley, 1996.
- [65] Z.J. Koles. The quantitative extraction and topographic mapping of the abnormal components in the clinical EEG. *Electroencephalography and Clinical Neurophysiology*, 79(6) :440–447, December 1991.
- [66] A. Kübler, A. Furdea, S. Halder, E.M. Hammer, F. Nijboer, and B. Kotchoubey. *A brain-computer interface controlled auditory event-related potential (p300) spelling system for locked-in patients*, volume 1157. 2009.
- [67] T.N. Lal, M. Schröder, T. Hinterberger, J. Weston, M. Bogdan, N. Birbaumer, and B. Schölkopf. Support vector channel selection in bci. *IEEE Trans. Biomed. Eng.*, 51(6) :1003–1010, 2004.

- [68] T. Lan, D. Erdogmus, A. Adami, M. Pavel, and S. Mathan. Salient EEG channel selection in brain computer interfaces by mutual information maximization. In *27th Annual International Conference of the Engineering in Medicine and Biology Society, 2005. IEEE-EMBS 2005.*, pages 7064–7067, 2005.
- [69] M. A. Lebedev and M. A. L. Nicolelis. Brain-machine interfaces : past, present and future. *Trends in Neurosciences*, 29(9) :536–46, September 2006.
- [70] P.-L. Lee, J.-J. Sie, Y.-J. Liu, C.-H. Wu, M.-H. Lee, C.-H. Shu, P.-H. Li, C.-W. Sun, and K.-K. Shyu. An SSVEP-actuated brain computer interface using phase-tagged flickering sequences : A cursor system. *Annals of Biomedical Engineering*, 38(7) :2383–2397, 2010.
- [71] S. Lemm, B. Blankertz, G. Curio, and K. R. Müller. Spatio-spectral filters for improving the classification of single trial EEG. *IEEE Trans. Biomed. Eng.*, 52(9) :1541–1548, 2005.
- [72] Y. Li. *Multichannel EEG signal classification - A geometric approach*. PhD thesis, McMaster University, 2011.
- [73] Y. Li, K.M. Wong, and H. deBruin. EEG signal classification based on a riemannian distance measure. In *IEEE Toronto International Conference Science and Technology for Humanity (TIC-STH)*, pages 268–273, 2009.
- [74] F. Lotte, M. Congedo, A. Lécuyer, F. Lamarche, and B. Arnaldi. A review of classification algorithms for EEG-based brain-computer interfaces. *Journal of Neural Engineering*, 4, 2007.
- [75] F. Lotte and G. Cuntai. Regularizing common spatial patterns to improve BCI designs : Unified theory and new algorithms. *IEEE Trans. Biomed. Eng.*, 58(2) :355–362, 2011.
- [76] H. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos. Regularized common spatial patterns with generic learning for EEG signal classification. 2009 :6599–6602, 2009.
- [77] T. Luth, D. Ojdanic, O. Friman, O. Prenzel, and A. Graser. Low level control in a semi-autonomous rehabilitation robotic system via a Brain-Computer interface. In *IEEE 10th International Conference on Rehabilitation Robotics.*, pages 721–728, 2007.
- [78] Jun Lv and Meichun Liu. Common spatial pattern and particle swarm optimization for channel selection in BCI. In *3rd International Conference on Innovative Computing Information and Control, 2008. ICICIC '08.*, page 457, 2008.
- [79] C. Mandel, T. Lüth, T. Laue, T. Röfer, A. Gräser, and B. Krieg-Brückner. Navigating a smart wheelchair with a brain-computer interface interpreting steady-state visual evoked potentials. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1118–1125, 2009.
- [80] S.G. Mason and G.E. Birch. A brain-controlled switch for asynchronous control applications. *IEEE Trans. Biomed. Eng.*, 47(10) :1297–1307, 2000.
- [81] D. J. McFarland, D. J. Krusienski, W. A. Sarnacki, and J. R. Wolpaw. Emulation of computer mouse control with a noninvasive brain-computer interface. *Journal of Neural Engineering*, 5(2) :101–110, 2008.
- [82] D.J. McFarland, W.A. Sarnacki, and J.R. Wolpaw. Electroencephalographic (EEG) control of three-dimensional movement. *Journal of Neural Engineering*, 7(3), 2010.
- [83] J. del R. Millan, F Galan, D Vanhooydonck, E Lew, J Philips, and M Nuttin. Asynchronous non-invasive brain-actuated control of an intelligent wheelchair. *IEEE International Conference of the Engineering in Medicine and Biology Society*, 2009 :3361–3364, 2009.

- [84] J. del .R. Millan, F. Renkens, J. Mourino, and W. Gerstner. Noninvasive brain-actuated control of a mobile robot by human EEG. *IEEE Transactions on Biomedical Engineering*, 51(6) :1026–1033, 2004.
- [85] M. Moakher. A differential geometric approach to the geometric mean of symmetric Positive-Definite matrices. *SIAM J. Matrix Anal. Appl.*, 26(3) :735–747, 2005.
- [86] G.R. Muller-Putz, R. Scherer, G. Pfurtscheller, C. Neuper, and R. Rupp. Non-invasive control of neuroprostheses for the upper extremity : Temporal coding of brain patterns. In *IEEE International Conference of the Engineering in Medicine and Biology Society*, pages 3353–3356, 2009.
- [87] A.R. Murguialday, V. Aggarwal, A. Chatterjee, Yoonju Cho, R. Rasmussen, B. O'Rourke, S. Acharya, and N.V. Thakor. Brain-Computer interface for a prosthetic hand using local machine control and haptic feedback. In *10th IEEE International Conference on Rehabilitation Robotics*, pages 609–613, 2007.
- [88] G. Müller-Putz, V. Kaiser, T. Solis-Escalante, and G. Pfurtscheller. Fast set-up asynchronous brain-switch based on detection of foot motor imagery in 1-channel EEG. *Medical and Biological Engineering and Computing*, 48(3) :229–233, March 2010.
- [89] G.R. Müller-Putz, R. Scherer, C. Neuper, and G. Pfurtscheller. Steady-state somatosensory evoked potentials : Suitable brain signals for brain-computer interfaces ? *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 14(1) :30–37, 2006.
- [90] G. Pfurtscheller. The hybrid BCI. *Frontiers in Neuroscience*, 2010.
- [91] G. Pfurtscheller, C. Brunner, A. Schlögl, and F.H. Lopes da Silva. Mu rhythm (de)synchronization and EEG single-trial classification of different motor imagery tasks. *NeuroImage*, 31(1) :153–159, May 2006.
- [92] G. Pfurtscheller and F.H. Lopes da Silva. Event-related EEG/MEG synchronization and desynchronization : basic principles. *Clinical Neurophysiology*, 110(11) :1842–1857, November 1999.
- [93] G Pfurtscheller and C Neuper. Motor imagery and direct brain-computer communication. *Proceedings of the IEEE*, 89(7) :1123–1134, 2002.
- [94] G. Pfurtscheller and T. Solis-Escalante. Could the beta rebound in the EEG be suitable to realize a "brain switch" ? *Clinical Neurophysiology*, 120(1) :24–29, January 2009.
- [95] G. Pfurtscheller, T. Solis-Escalante, R. Ortner, P. Linortner, and G. R. Muller-Putz. Self-Paced operation of an SSVEP-Based orthosis with and without an Imagery-Based "Brain switch : " a feasibility study towards a hybrid BCI. *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, 18 :409–414, 2010.
- [96] G. Pfurtscheller, T. Solis-Escalante, R. Ortner, P. Linortner, and G. R. Müller-Putz. Self-Paced operation of an SSVEP-Based orthosis with and without an Imagery-Based Brain switch : A feasibility study towards a hybrid BCI. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 18(4) :409–414, 2010.
- [97] D. T. Pham. Joint approximate diagonalization of positive definite hermitian matrices. *SIAM Journal on Matrix Analysis and Applications*, 22 :1136, 2001.
- [98] D.T. Pham and J.F. Cardoso. Blind separation of instantaneous mixtures of nonstationary sources. *IEEE Transactions on Signal Processing*, 49(9) :1837–1848, 2001.
- [99] J. Philips, J. del R. Millan, G. Vanacker, E. Lew, F. Galan, P.W. Ferrez, H. Van Brussel, and M. Nuttin. Adaptive shared control of a Brain-Actuated simulated wheelchair. In *IEEE 10th International Conference on Rehabilitation Robotics*, pages 408–414, 2007.

- [100] G. Pires, M. Castelo-Branco, and U. Nunes. Visual p300-based BCI to steer a wheelchair : A bayesian approach. In *30th Annual IEEE International Conference of the Engineering in Medicine and Biology Society, EMBS*, pages 658–661, 2008.
- [101] J. Polich. Updating p300 : An integrative theory of p3a and p3b. *Clinical Neurophysiology*, 118(10) :2128–2148, 2007.
- [102] F. Porikli, P. Meer, and O. Tuzel. Human detection via classification on riemannian manifolds. *Conference on computer vision and pattern recognition*, pages 1–8, 2007.
- [103] K. Qian, P. Nikolov, D. Huang, D. Y. Fei, X. Chen, and O. Bai. A motor imagery-based online interactive brain-controlled switch : Paradigm development and preliminary test. *Clinical Neurophysiology*, In Press, Corrected Proof, 2010.
- [104] H. Ramoser, J. Muller-Gerking, and G. Pfurtscheller. Optimal spatial filtering of single trial EEG during imagined hand movement. *IEEE Transactions on Rehabilitation Engineering*, 8(4) :441–446, December 2000.
- [105] B. Rebsamen, E. Burdet, G. Cuntai, C. L. Teo, Q. Zeng, M. Ang, and C. Laugier. Controlling a wheelchair using a BCI with low information transfer rate. In *10th IEEE International Conference on Rehabilitation Robotics, ICORR*, pages 1003–1008, 2007.
- [106] B. Rebsamen, E. Burdet, G. Cuntai, H. Zhang, C. L. Teo, Q. Zeng, M. Ang, and C. Laugier. A Brain-Controlled wheelchair based on p300 and path guidance. In *The First IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics*, pages 1101–1106, 2006.
- [107] B. Rivet, A. Souloumiac, V. Attina, and G. Gibert. xDAWN algorithm to enhance evoked potentials : application to brain-computer interface. *IEEE Transactions on Bio-Medical Engineering*, 56(8) :2035–2043, August 2009.
- [108] P. J. Rousseeuw and K. Van Driessen. A fast algorithm for the minimum covariance determinant estimator. *Technometrics*, 41 :212–223, 1999.
- [109] C. Sannelli, T. Dickhaus, S. Halder, E. M. Hammer, K. R. Müller, and B. Blankertz. On optimal channel configurations for SMR-based Brain-Computer interfaces. *Brain Topography*.
- [110] A. Schlögl, F. Lee, H. Bischof, and G. Pfurtscheller. Characterization of four-class motor imagery eeg data for the bci-competition 2005. *Journal of Neural Engineering*, 2(4), 2005.
- [111] J. Schäfer and K. Strimmer. A shrinkage approach to large-scale covariance matrix estimation and implications for functional genomics. *Statistical Applications in Genetics and Molecular Biology*, 4, 2005.
- [112] L. T. Skovgaard. A riemannian geometry of the multivariate normal model. *Scandinavian Journal of Statistics*, 11(4) :211–223, 1984.
- [113] H. Snoussi and C. Richard. Monte carlo tracking on the riemannian manifold of multivariate normal distributions. In *IEEE 13th Digital Signal Processing Workshop and 5th IEEE Signal Processing Education Workshop DSP/SPE*, pages 280–285. IEEE, January 2009.
- [114] M. Tenenhaus. *Statistique : Méthodes pour décrire, expliquer et prévoir*. Dunod, 2e édition edition, December 2006.
- [115] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society.*, 58(1) :267–288, 1996.
- [116] O. Tuzel, F. Porikli, and P. Meer. Pedestrian detection via classification on Riemannian manifolds. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(10) :1713–1727, 2008.

- [117] D. Valbuena, M. Cyriacks, O. Friman, I. Volosyak, and A. Graser. Brain-Computer interface for high-level control of rehabilitation robotic systems. In *IEEE 10th International Conference on Rehabilitation Robotics*, pages 619–625, 2007.
- [118] F.B. Vialatte, M. Maurice, J. Dauwels, and A. Cichocki. Steady-state visually evoked potentials : Focus on essential paradigms and future perspectives. *Progress in Neurobiology*, In Press, Corrected Proof, 2010.
- [119] J. J. Vidal. Toward direct brain-computer communication. *Annual Review of Biophysics and Bioengineering*, 2 :157–180, 1973.
- [120] J. J. Vidal. Real-time detection of brain events in EEG. *Proceedings of the IEEE*, 65(5) :641–633, 1977.
- [121] C. Vidaurre and B. Blankertz. Towards a cure for BCI illiteracy. *Brain Topography*, 23(2) :194–198, 2010.
- [122] A. Villringer, J. Planck, C. Hock, L. Schleinkofer, and U. Dirnagl. Near infrared spectroscopy (NIRS) : a new tool to study hemodynamic changes during activation of brain function in human adults. *Neuroscience Letters*, 154(1-2) :101–104, May 1993.
- [123] H. Wang. Optimizing spatial filters for single-trial EEG classification via a discriminant extension to CSP : the fisher criterion. *Medical & Biological Engineering & Computing*, 49(9) :997–1001, 2011.
- [124] H. Wang and W. Zheng. Local temporal common spatial patterns for robust Single-Trial EEG classification. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 16(2) :131–139, 2008.
- [125] Y. Wang, S. Gao, and X. Gao. Common spatial pattern method for channel selection in motor imagery based brain-computer interface. In *Engineering in Medicine and Biology Society, 2005. IEEE-EMBS 2005. 27th Annual International Conference of the*, pages 5392–5395, 2005.
- [126] J. R. Wolpaw. Two-dimensional movement control using electrocorticographic signals in humans. *Journal of Neural Engineering*, 5(1) :75–84, February 2008.
- [127] J. R. Wolpaw, N. Birbaumer, D. J. McFarland, G. Pfurtscheller, and T. M. Vaughan. Brain-computer interfaces for communication and control. *Clinical Neurophysiology*, 113(6) :767–791, 2002.
- [128] J. R. Wolpaw and D. J. McFarland. Control of a two-dimensional movement signal by a noninvasive brain-computer interface in humans. *Proceedings of the National Academy of Sciences of the United States of America*, 101(51) :17849–17854, December 2004.
- [129] J. R. Wolpaw, D. J. McFarland, G. W. Neat, and C. A. Forneris. An EEG-based brain-computer interface for cursor control. *Electroencephalography and Clinical Neurophysiology*, 78(3) :252–259, 1991.
- [130] Y. Xinyi, R. K. Ward, and G. E. Birch. Robust common spatial patterns for EEG signal preprocessing. In *30th Annual International IEEE Conference of the Engineering in Medicine and Biology Society*, pages 2087–2090, 2008.
- [131] Y. Xinyi, R. K. Ward, and G. E. Birch. Sparse spatial filter optimization for EEG channel reduction in brain-computer interface. In *IEEE International Conference on Acoustics, Speech and Signal Processing, 2008. ICASSP 2008*, pages 417–420, 2008.
- [132] Q. Zhao, L. Zhang, A. Cichocki, and J. Li. Incremental common spatial pattern algorithm for BCI. pages 2656–2659. *IEEE International Joint Conference on Neural Networks*, 2008.
- [133] Y. Zhou, S.G. Mason, and G.E. Birch. Enhancing the performance of the LF-ASD brain-computer interface. volume 3, pages 2443–2444 vol.3, 2002.

- [134] H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society*, 67 :301–320, 2005.